Applications in Agent-Based Computational Economics

Stephan Schuster

University of Surrey

January 2012

Applications in Agent-Based Computational Economics

Stephan Schuster

Thesis submitted for the degree of Doctor of Philosophy

Department of Economics, Faculty of Arts and Human Sciences
University of Surrey

January 2012

Abstract

A constituent feature of adaptive complex systems are non-linear feedback mechanisms between actors. This makes it often difficult to model and analyse them. Agent-based Computational Economics (ACE) uses computer simulation methods to represent such systems and analyse non-linear processes.

The aim of this thesis is to explore ways of modelling adaptive agents in ACE models. Its major contribution is of a methodological nature. Artificial intelligence and machine learning methods are used to represent agents and learning processes.

In this work, a general reinforcement learning framework is developed and realised in a simulation system. This system is used to implement three models of increasing complexity in two different economic domains. One of these domains are iterative games in which agents meet repeatedly and interact. In an experimental labour market, it is shown how statistical discrimination can be generated simply by the learning algorithm used. The results resemble actual patterns of observed human behaviour in laboratory settings. The second model treats strategic network formation. The main contribution here is to show how agent-based modelling helps to analyse non-linearity that is introduced when assumptions of perfect information

and full rationality are relaxed. The other domain has a Health Economics background. The aim here is to provide insights of how the approach might be useful in real-world applications. For this, a general model of primary care is developed, and the implications of different consumer behaviour (based on the learning features introduced before) analysed.

Contents

Contents i

List of Symbols and Abbreviations iv

List of Figures v

List of Tables viii

1 Introduction 1

2 A Computational Framework for Modelling Learning 8
- 2.1 Introduction . . . 8
- 2.2 Experience-based Learning . . . 11
  - 2.2.1 Experimental Games Using Simple RL Models . . . 15
  - 2.2.2 Experimental Games Using Combined Belief and RL models . . . 18
  - 2.2.3 Analytical Approaches with Simple RL Models . . . 21
  - 2.2.4 Cognitive Approaches . . . 26
- 2.3 Concept . . . 31
- 2.4 The Algorithm . . . 34
  - 2.4.1 Reinforcement Learning . . . 35
  - 2.4.2 State Space Partitioning . . . 36
  - 2.4.3 The Complete Algorithm . . . 46
  - 2.4.4 Compact Notation . . . 48
- 2.5 Relation to Existing Approaches . . . 50
- 2.6 An Example . . . 52
- 2.7 Conclusion and Outlook . . . 55

3 Statistical Discrimination 59
- 3.1 Introduction . . . 59
- 3.2 Models of Statistical Discrimination . . . 62

3.3 Experiments with Statistical Discrimination 74
3.4 A Reinforcement Learning Model of Statistical Discrimination 77
3.5 Simulations 83
3.5.1 Exploration 84
3.5.1.1 Finding Optimal Learning Parameters . . . 84
3.5.1.2 Variant I 88
3.5.1.3 Variant II 92
3.5.1.4 Variant III 94
3.5.2 Average Results 94
3.5.2.1 Variant I 98
3.5.2.2 Variant II 101
3.5.3 How Persistent is Discrimination? 106
3.5.4 Summary of the Simulation Results 116
3.6 Conclusion 118

4 Network Formation 120
4.1 Introduction 120
4.2 Definitions and Notation 125
4.2.1 Graphs 125
4.2.2 Games on Graphs 127
4.2.3 Stability definitions 128
4.3 Models of Network Formation 129
4.4 Experiments with Network Formation 138
4.5 A Reinforcement Learning Model of Network Formation . . 144
4.6 Simulations 148
4.6.1 Overview 148
4.6.2 Network Properties for Different α 150
4.6.3 Network Structure and Dynamics 153
4.6.4 Memory Effects 158
4.6.5 Summary of the Simulation Results 159
4.7 Applying BRA 162
4.8 Comparison with Empirical Results 166
4.9 Conclusion 171

5 The Market for Primary Care 174
5.1 Introduction 174
5.2 Background 176
5.3 GP Behaviour 178
5.4 Patient Behaviour 189
5.5 Modelling Primary Care 192
5.6 A Reinforcement Learning Model Of Primary Care 194
5.6.1 Overview 195
5.6.2 Patient Decisions 196

5.6.3 GP Decisions . . . 198
5.7 Simulations . . . 206
5.7.1 Exploration of the Model . . . 206
5.7.1.1 Parameter Settings and Setup . . . 206
5.7.1.2 Static Analysis . . . 211
5.7.2 Dynamic Analysis . . . 217
5.7.3 Summary of the Simulation Results and Discussion . 220
5.8 Conclusion . . . 225

6 Conclusion 227

Bibliography 233

Appendices 251

A A Scalable ACE Simulation Software Framework 252
A.1 Introduction . . . 252
A.2 Model Representation System . . . 254
A.2.1 The Frame Principle . . . 254
A.2.2 Formal Description as Language . . . 257
A.2.3 Agent Behaviour . . . 263
A.2.4 Agent Communication . . . 269
A.2.5 Other Components . . . 270
A.2.6 Interfaces . . . 270
A.3 Software Architecture . . . 275
A.3.1 Base system . . . 275
A.3.1.1 Architecture and Design . . . 275
A.3.1.2 Implementation . . . 278
A.3.2 Distributed system . . . 278
A.3.2.1 Basic Design Questions . . . 278
A.3.2.2 Architecture and Design . . . 281
A.3.2.3 Implementation . . . 283
A.3.2.4 Examples . . . 288
A.4 Conclusion . . . 290

B Details of the Statistical Discrimination Model 293

C Variance Analysis for the Primary Care Model 296

D Software and Models 302

List of Symbols and Abbreviations

Abbreviation	Description	Definition
ACE	Agent-Based Economics	page 26
ABM	Agent-Based Modelling	page 28
ACS	Anticipation-based Classifier System	page 28
AI	Artificial Intelligence	page 254
AL	Aspiration level	page 24
API	Application Programming Interface	page 252
BRA	Bounded Rationality Algorithm	page 34
BM	Bush-Mosteller model	page 12
CL	Coate/Loury model	page 62
CBR	Case Based Reasoning	page 24
EWA	Experience Weighted Attraction Learning	page 18
EJB	Enterprise Java Beans	page 285
ER	Erev-Roth	page 15
FFS	Fee for service	page 180
GA	Genetic Algorithm	page 84
GP	General Practitioner	page 176
gsim	Generic simulation framework	page 252
HA	Health Authority	page 194
JEE	Java Enterprise Edition	page 283
JMS	Java Messaging System	page 285
LCS	Learning Classifier System	page 28
PA	Payoff assessment model	page 16
RL	Reinforcement learning	page 2
RMI	Remote Method Invocation Protocol	page 285

List of Figures

2.1 BRA - representation of the state space 39
2.2 BRA - expansion process principle 42
2.3 BRA - generalisation principle 43
2.4 BRA - switching principle 45
2.5 BRA - example 1 54
2.6 BRA - example 2 54

3.1 Equilibrium in Coate and Loury's statistical discrimination model. 67
3.2 Statistical discrimination - model fit model variant I 88
3.3 Statistical discrimination - model fit in model variant II 89
3.4 Statistical discrimination - model fit in model variant III 89
3.5 Statistical discrimination - hiring rate model variant I 90
3.6 Statistical discrimination - investment rate model variant I . . . 91
3.7 Statistical discrimination - hiring rate model variant II 93
3.8 Statistical discrimination - investment rate model variant II . . 93
3.9 Statistical discrimination - hiring rate model variant III 95
3.10 Statistical discrimination - investment rate model III 95
3.11 Statistical discrimination - average hiring rates for various signal probabilities in model variant I 99
3.12 Statistical discrimination - hiring rates for various signal probabilities in model variant I 99
3.13 Statistical discrimination - sample simulation runs for variant I 100
3.14 Statistical discrimination - average hiring rates for various signal probabilities in model variant II 103
3.15 Statistical discrimination - hiring rates for various signal probabilities in model variant II 103
3.16 Statistical discrimination - sample simulations for variant II . . 104
3.17 Statistical discrimination - hiring rates if firms deterministically discriminate (green group) 109

3.18 Statistical discrimination - hiring rates if firms deterministically discriminate (purple group) . . . 110
3.19 Statistical discrimination - Effect of cost heterogeneity in model variant II . . . 111
3.20 Statistical discrimination - Effect of ex ante beliefs in model variant II . . . 112
3.21 Statistical discrimination - discrimination of green workers for various $\delta_{f(\theta)}$ and ex ante beliefs . . . 116

4.1 Network formation - model fit for various parameters . . . 148
4.2 Network formation - network density for all cost samples . . . 150
4.3 Network formation - density, stability and fit for the low cost range 151
4.4 Network formation - density, stability and fit for the medium cost range . . . 152
4.5 Network formation - density, stability and fit for the high cost range . . . 153
4.6 Network formation - density, stability and fit for the low cost range (BRA model) . . . 164
4.7 Network formation - rule extractions in the low cost range (BRA model) . . . 164
4.8 Network formation - rule extractions in the medium cost range (BRA model) . . . 165
4.9 Network formation - rule extractions in the high cost range (BRA model) . . . 165
4.10 Network formation - average payoff for various α and γ values in the simultaneous linking game. . . . 168
4.11 Network formation - density, stability and frequency of Nash networks over time in the simultaneous linking game. . . . 169

5.1 Primary care - GP utility function . . . 210
5.2 Primary care - waiting lists (static analysis) . . . 212
5.3 Primary care - referrals (static analysis) . . . 214
5.4 Primary care - GP effort (static analysis) . . . 215
5.5 Primary care - patient utility (static analysis) . . . 216
5.6 Primary care - waiting lists (dynamic analysis) . . . 218
5.7 Primary care - referrals (dynamic analysis) . . . 219
5.8 Primary care - GP effort (dynamic analysis) . . . 220
5.9 Primary care - patient utility (dynamic analysis) . . . 221

A.1 Knowledge representation in gsim. . . . 255
A.2 The base agent frame in gsim . . . 256
A.3 gsim base architecture . . . 277
A.4 gsim distributed architecture . . . 289

A.5 gsim simulation examples I 290
A.6 gsim simulation examples II 291

List of Tables

2.1 BRA - Summary of notation 46
2.2 BRA - payoffs of the demand game 53

3.1 Statistical discrimination - payoffs of the RL model 78
3.2 Statistical discrimination - description of model variant I 80
3.3 Statistical discrimination - description of model variant II . . . 81
3.4 Statistical discrimination - description of model variant III . . . 83
3.5 Statistical discrimination - simulation parameters for finding optimal RL parameters. 87
3.6 Statistical discrimination - simulation parameters for obtaining average results. 97
3.7 Statistical discrimination - rules in model variant I, sample 1 . . 101
3.8 Statistical discrimination - rules in model variant I, sample 2 . . 102
3.9 Statistical discrimination - rules in model variant II, sample 1 . 106
3.10 Statistical discrimination - rules in model variant II, sample 2 . 107
3.11 Statistical Discrimination - rules in model variant II, sample 1 . 114
3.12 Statistical Discrimination - rules in model variant II, sample 1 . 115

4.1 Network formation - RL model parameter settings 146
4.2 Network formation - low cost range network structures 155
4.3 Network formation - medium cost range network structures . . . 156
4.4 Network formation - high cost range network structures 157
4.5 Network formation - network measures for different γ values . . 158
4.6 Network formation - RL model parameter settings for the simultaneous linking game 167
4.7 Network formation - Comparison of payoffs of equilibrium prediction, experimental and simulated results in the simultaneous linking game. 168
4.8 Network formation - Nash networks visited in the simultaneous linking game. 170

5.1 Primary care - overview of simulation runs 209
5.2 Primary care - utility functions 209
5.3 Primary care - mean and standard deviation of health outcomes (dynamic analysis) 222
5.4 Primary care - patient loyalty (dynamic analysis) 224

B.1 Statistical Discrimination - average discrimination in model variant I 294
B.2 Statistical Discrimination - average discrimination in model variant II 295

D.1 List of executable models 303
D.2 List of source code 303

Chapter 1

Introduction

Economies can be seen as complex dynamics systems: Many autonomous agents interact locally, giving rise to global phenomena such as price levels, growth rates, etc. As Tesfatsion (2006) notes, the study of these macro phenomena require strong abstractions and simplifications, which, if removed, quickly make the system intractable. For example, what would happen if the Walrasian Auctioneer would be removed in a standard Walrasian model? Because of this 'small' perturbation, the modeller now has to 'come to grips with challenging issues such as asymmetric information, strategic interaction, expectation formation on the basis of limited information, mutual learning, social norms, transaction costs, externalities, market power, predation, collusion, and the possibility of coordination failure (convergence to a Pareto-dominated equilibrium)' (Tesfatsion 2006). Agent-based computational economics (ACE) is a method that has emerged as a novel way to look at the evolution of such equilibria and global phenomena by generating, or 'growing' them endogenously (Epstein and Axtell 1996). It is a way to computationally study artificial worlds modelled as dynamic systems of interacting entities. The entities are typically individuals or social

groups such as consumers, firms or players in games. Furthermore, physical entities such as infrastructure or spatial settings might be represented in a computational model. Models are analysed by simulating them in a computer, and interpreting the results that are generated.

A system is called complex if it is composed of interacting units and if has emergent properties, that is, properties arising from the interactions of the agents. Following Tesfatsion (2006), a system is complex adaptive if the units of the system have some form of pro- and reactive capabilities. There are basically three definitions of complex adaptive systems:

Definition 1. *A complex adaptive system is a complex system that includes reactive units, i.e., units capable of exhibiting systematically different attributes in reaction to changed environmental conditions.*

Definition 2. *A complex adaptive system is a complex system that includes goal-directed units, i.e., units that are reactive and that direct at least some of their reactions towards the achievement of built-in (or evolved) goals.*

Definition 3. *A complex adaptive system is a complex system that includes planner units, i.e., units that are goal-directed and that attempt to exert some degree of control over their environment to facilitate achievement of these goals.*

Essentially, economic systems can be defined as complex adaptive systems, composed of intelligent agents. Some form of cognition and goal-directedness is essential to most models. However, the degree of goal-direction and cognitive capabilities of agents varies strongly. The simplest models, for example, represent only reactions to neighbouring agents' states (e.g. Schelling 1971). In game theory, simple reinforcement learning (RL), as well as mixed systems, combining cognitive learning mechanisms with

experience-based learning, have been widely applied (for details, see chapter 3; for an overview see Brenner (2006)). There is no simple rule which models should use which sort of learning; this typically depends on the nature of the domain. For example, in environments where habitualisation is a prominent feature, e.g. in repeated game situations, simple reinforcement learning matches actual behaviour usually reasonably well. On the other hand, if decisions are less frequent and more important, simple learning mechanisms are not accurate representations. For instance, it could be argued that choosing a doctor (see chapter 4) is a very conscious decision, thus RL would be inappropriate and some mechanism for representing beliefs and judgements would be the more natural choice.

While the role of ACE as a tool for simulating complex systems is straightforward, its role as a paradigm for economic modelling is controversial. Typical criticisms of ACE models regard the following points (e.g. Fagiolo et al 2007; Leombruni and Richiardi 2005; Richiardi 2003):

- The lack of standardisation and formalism of ACE models. The sheer mass and heterogeneity of models makes unclear what this approach actually stands for. In general, there are almost no standardised techniques to analyse agent-based models, for example, whether and when sensitivity analyses should be conducted, how timing should be interpreted and so on.

- The lack or impossibility of empirical validation of many models. Many simulations use some stylised facts to establish the validity of the model. Calibration is typically an iterative process where the modeller reduces the parameter space to smaller ranges which generate the most plausible results, or where detailed data exists, to a dataset. However, since one of the advantages of ACE models is the

integration of more 'realism' in the form of exact agent specifications, there is always a trade-off between descriptive accuracy and analytical tractability. Naturally, the more degrees of freedom a model has, the more difficult it is to map it to available empirical evidence (due to the number of parameters to calibrate).

– The lack of generality and unclear approach to handle results. Whereas it is straightforward to estimate, say, reduced forms, or calculate transition probabilities on empirical data, artificial data can only be calibrated against some empirical benchmark. A result derived from artificial data can only be as good as the underlying simulation is able to replicate the actual real-world process. Furthermore, agent-based models are likely to underidentify actual trends. ACE models are richer, and therefore, create more noise. Another aspect of this problem is 'equifinality'. Equifinality describes the case when a number of different models may generate similar data, that is, they may equally well explain the same phenomenon but by different processes.

Some ACE modellers view agent-based modelling as a new way of doing science (Epstein and Axtell 1996). The main interest of researchers in this area is to discover new rules, theories and test hypotheses about the *processes* that generate certain phenomena, and only later derive analytical better models that explain larger classes of phenomena (e.g. Edmonds and Moss 2005). As these modellers typically use their simulations on a mere qualitative basis, as thought-experiments or support for generating new hypotheses, there is no rationale for testing such models against empirical data. Although immunised against empirical falsification, some forecasting exercises might still be possible, but results have to be treated with caution. More importantly, there is a danger that 'one ends up building

auto-referential formalisations that have no link to reality' (e.g. Fagiolo et al 2007).

The aim of this thesis is to apply RL methods as a means to model adaptive feedback processes. The overall contribution is of a methodological nature. The models presented have the main purpose to demonstrate this method and show how it can be applied to a range of problems. In that sense, the models discussed in the thesis fall into the last category of models: They are mainly of a qualitative nature; empirical validation is not the main interest of the simulations.

The focus of this thesis is reinforcement learning. Reinforcement learning is a very simple experience based learning approach; agents learn by trial and error. It has often been used in the ACE literature, but often ad-hoc or in simple models. Moreover, there are only few approaches which integrate experience-based learning with cognitive elements such as beliefs.

The objectives of the thesis are to

1. Develop a new computational approach that integrates RL with simple cognitive elements. It shall provide a new approach of modelling human decision processes.

2. Apply RL to economic, mainly game-theory models and contribute to the learning literature in this field. As the use of simulations allows to build more complex models, an important aspect of this thesis is to build a 'bridge' between pure game theory and empirical results of experimental game theory. A recurring topic is therefore the comparison with experimental evidence.

3. Analyse the impact of different learning approaches in more complex

domains. Here, the question is how RL can be used to enrich the analysis of more applied, real-world models.

As the methodological basis, chapter 2 reviews RL in the economic literature and develops a general learning framework, combining reinforcement and rule learning. The motivation is to provide an alternative, generic way of representing agent decision mechanisms in a unified framework for several classes of models. It tries to go beyond simplistic fomalisations of adaptive capabilities such as simple RL, but to keep computational complexity within bounds. Chapter 3 applies this approach to a model of statistical discrimination. It is shown that the framework is capable of reproducing patterns of actual human behaviour in game-theoretic experiments. Chapter 4 is an application of RL to network formation. Results of the learning process are compared with axiomatic results for perfectly rational players. A modified version of the model is then used to reproduce an experiment and to compare its behaviour with observed human behaviour. A very different model is presented in chapter 5. While the purpose of the first chapters is to apply and analyse learning in rather simple settings, the purpose of this chapter is to use it in a complex setting with many influencing variables. The requirements for adaptation in this application are very different from that discussed before: In the model, doctors decide about treatment patterns, quality and their own workload. Patients choose doctors based on their own experience and recommendations of other consumers. Several simulations using different learning and choice scenarios are compared.

The models have been implemented in their own software framework, providing the learning features used in the thesis. Appendix A describes the architecture and implementation of the software.

This work contributes in several ways to the ACE literature:

– It adds a novel algorithm for representing learning in artificial agents. This approach has been published in Schuster (2012).

– It applies RL to statistical discrimination games. It belongs thus to the few dynamic models in this area, and is to the knowledge of the author the first using an RL approach.

– It applies RL to strategic network formation games. So far, adaptation in the strategic network formation literature has received almost no attention. Here, adaptation is applied for the first time to the well-known connections model of Jackson and Wolinsky (1996).

– It provides one of the first applications of ACE in the field of health care system modelling. So far, only few agent-based models in this area have been proposed, and in fact, there has been no ACE model of primary care.

Chapter 2

A Computational Framework for Modelling Learning

2.1 Introduction

The perfectly informed and rational homo oeconomicus has often been criticised as too unrealistic - humans would not have the computational power to calculate the best decisions, taking into account all information and all possible outcomes. Already Simon (Simon 1956b) argued to use simpler, psychologically more plausible algorithms. While the argument of bounded rationality is frequently used as critique of the standard economic model, the argument remains, however, vague (Simon 2000) - meaning usually everything that is not classical economics, ranging, for example, from systematic errors people make in judgements to the research on decision heuristics as an alternative form of decision making.

Common to all critiques of perfect rationality is that humans are not capable of doing the computations required by a homo oeconomicus, but are bound to commit errors and misjudgements. As some psychologists (e.g.

Gigerenzer and Goldstein 1996; Lopes 1994) point out, most alternative models are still based on the fundamental assumption that expected utility and Bayesian reasoning are the basis for all human decision making under uncertainty. For example, subjective expected utility theory acknowledged that individuals are not fully informed, and replaced objective probabilities with subjective; however, the basis for reasoning remained the same.

In the sociological and psychological literature, a vast amount of evidence has been collected to show experimentally how this classical model can fail. Formalisation, however, is rare. An example is Prospect Theory (Kahnemann and Tversky 1979). The main argument of Prospect Theory is that people value future losses more highly than potential gains. Prospect theory proposes an S-shaped value function that is concave for gains, and convex for losses. That is, individuals become risk avoiding the higher the potential losses, and risk seeking the greater the potential gains. Another aspect of the value function has been characterised by loss aversion, which is usually represented by a steeper slope of the curve in the loss area. These aspects have been used to explain apparently irrational, as well as loss avoiding behaviour in many psychological experiments. Psychologists have also emphasised that humans process information not as the Bayesian paradigm postulates, but rather crudely by using decision heuristics and cues from their environment. In the field of cognitive psychology bounded rationality became almost exclusively associated with this perspective in cognitive psychology. The behavioural aspect of bounded rationality (like learning by doing) has been neglected or not seen as a subject for this discipline (Gigerenzer and Goldstein 1996).

In the economic literature, the most common way of modelling bounded rationality is to postulate deviations from perfect rationality - for example, by introducing an error term or some random noise (Auman 1997).

With RL approaches a more behavioural dimension has become available in (behavioural) game theory. In pure stimulus-response models, agents learn by trial and error without any explicit knowledge representation (e.g Roth and Erev 1995). Some authors combine experience learning with foresight in mixed models as in fictitious play (Camerer and Ho 1999). Some ACE models are based on similar concepts (see Brenner (2006) for an overview); especially classifier systems have received interest to represent a simple form of rule learning (e.g. Kirman and Vriend 2001a; LeBaron et al 1999).

Another angle of decision making can be seen in cognitive architectures. Architectures such as ACT-R (e.g. Anderson 1993) and Soar (e.g. Lehman et al 2003) try to simulate human decision-making as a computer program. Most of them focus on the working of the mind when solving, say, mathematical problems and model in detail what processing steps are involved in solving such problems. More recently, Sun showed how his cognitive architecture CLARION (Sun and Slusarz 2005) can be connected with social simulation. In this approach, the environment of an organism can, in contrast to the classical architectures, be represented in an agent's mind (Sun and Naveh 2007).

In this chapter, a computational model of bounded rationality is developed that addresses the tension between simplifying representations as pure stimulus-response learning on the one end of the spectrum, and often complex higher levels of cognition on the other end. It is most closely related to mixed models and classifier systems, and has analogies with Sun's application of CLARION. However, there is no distinct social or economic approach to individual learning. The algorithm described in this paper attempts to fill this gap.

In the remainder of the chapter, the related literature is reviewed in

some detail. Then, a simple conceptual framework based on Simon's concept of bounded rationality (Simon 1956b) is described, before outlining concept and algorithm in more detail. The algorithm is then related to the existing approaches in the literature. A simple simulation illustrates how the algorithm works. The conclusion also outlines how the framework is related to the learning problems in the applications in chapters 3 to 5.

2.2 Experience-based Learning

Humans learn through a variety of sources, such as own experience, observation, imitation or cognition. According to Brenner (2006), learning in Economic models can be distinguished according to the degree of consciousness in decisions. On the one end of the spectrum, humans learn in a very simple way by reacting to stimuli. This type of learning happens automatically on an unconscious level; in routine situations, humans are often incapable of explaining why they are doing things in a certain way. On the other end, learning happens in a conscious way by reflecting, e.g. about own experiences or about observations. Actions resulting from such deliberation originate from the mental model humans have about the world, and is disconnected from immediate stimuli. In between, there are several modes of learning, which can be characterised as routine learning. They have in common that they usually use some kind of experience. Brenner subsumes many kinds of learning under experience-based learning: Reinforcement learning, learning by imitation, satisficing (searching for satisfactory problem solutions) or collecting and analysing experience. Fictitious play, a common learning technique in game theory, is the typical example for the latter. In fictitious play, players remember their payoffs and strategies and compare them with payoffs and strategies of other players in the game. Using this information, they compute what they would have earned if they played the

other strategies. If the other strategies fare better, the player can then switch his behaviour. While experience is necessary to learn in fictitious play, it requires also a cognitive component, namely the reflection upon other players' actions. Pure belief-based approaches do not use the feedback coming from own activities. A typical example is Bayesian learning, which updates beliefs about future states an agent will be in. Cognitive architectures from Psychology can be seen as a similar example. These approaches aim to model mental processes in the brain, and as such are typically independent of concrete experience.

The aim of this chapter is to develop an algorithm that can be applied to a wide range of ACE modelling problems. Thus, approaches that do not require prior knowledge about the domain are the most relevant. Experience-based learning methods are a natural candidate for this, since they acquire knowledge incrementally and base decisions on that knowledge. The literature reviewed here looks therefore mainly at experience learning, in particular reinforcement learning, but not pure belief-based learning. Furthermore, throughout the thesis, RL will be used as a synonym for any experience-based learning method that is based on RL.

In RL, agents learn to choose actions that were successful in the past more often, while they avoid actions that led to unsatisfactory outcomes. This is referred to as the 'Law of effect'. A basic learning model was first formalised by Bush and Mosteller (BM) (Bush and Mosteller 1955). According to BM, the choice probabilities p of an action at a given time can be computed according to

$$Qp = p + a(1 - p) - bp \tag{2.1}$$

where $a, 0 \leq a \leq 1$ describes rewards, and $b, 0 \leq b \leq 1$ punishments. Q

is a mathematical operator that describes the new quantity of p after the reward is applied. It is a short form to describe the stepwise update of reinforcements. Most learning models generalise the BM idea to a time-discounted version. The main components typically are:

- An action set A from which an action a is chosen, and payoffs π associated with them;
- An action strength function that updates the experience over time. The typical function is introduced in Roth and Erev (1995):

$$q_k(t+1) = q_k(t) + \pi(t) \tag{2.2}$$

  which updates the strength q of the k-th action with the current payoff (Roth and Erev 1995).

- A selection function that selects successful actions based on the q_k. This selection function is usually based on Luce's choice theorem (Luce 1959):

$$p_k = \frac{q_k}{\sum q_j} \tag{2.3}$$

  This function computes the choice probability of action k relative to its strength q_k.

Thus, BM-type models accumulate experience. There exist several problems with this simple type of learning. For example, after long periods of playing a single action, the learner will react to a change in payoffs extremely slowly, and hence possibly play inferior actions. On the other hand, if the learner reacts reasonably fast, it might be that it never locks in into optimal choices. Several RL models have addressed these problems differently. They can roughly be characterised as follows:

– Cumulative RL without aspirations (Roth and Erev 1995; Erev and Roth 1998; Laslier et al 2001; Laslier and Walliser 2005; Beggs 2005; Rustichini 1999; Camerer and Ho 1999), which are all based on the original BM model described above. Many analytical approaches use the simple version in combination with simple decision problems where adjustment to changing environments does not play a role; the problem does not exist in this case. Other models, mainly of are more empirical nature, vary the base model by adding forgetting and experimentation parameters (Erev and Roth 1998) or simple beliefs (Camerer and Ho 1999) to counterbalance the effect of excessive cumulation.

– Averaging mechanisms (Karandikar et al 1998; Mookherjee and Sopher 1994; 1997; Sarin and Vahid 2001; Gilboa and Schmeidler 1996). In principle, average reinforcements can be interpreted as a form of belief learning, namely as an expected future reward. The advantage is that agents can adjust reasonably fast to changes in the environment.

– Aspiration level models with cumulative RL (e.g Boergers and Sarin 2000) or averaging mechanisms (e.g. Karandikar et al 1998; Bendor et al 2001b; Napel 2003; Gotts et al 2007); see also Bendor et al (2001a) for an overview. In models of this type, action strengths are updated with respect to the difference to an exogenously set or endogenously evolving aspiration level. If the payoff is below this level, the reward is subtracted, otherwise added. Some models base the calculation of action probabilities on the distance from the actual payoff to the aspiration level. The probability distributions that determine action selection can be skewed to choose an action with a probability close to 1 if the reward is above, or close to 0 if the reward is below the aspiration level. When payoffs decrease, agents tend to play strategies

proportional to their expected payoffs, thereby achieving a similar exploration effect once their environment changes and payoffs decrease. The advantage of this approach is that lock-in into optimal choices is supported, at the same time not being deterministic if payoffs fall below the aspiration level.

2.2.1 Experimental Games Using Simple RL Models

One main motivation of many models has been the search for learning rules that predict experimental data better than the standard equilibrium prediction under full information (e.g. Roth and Erev 1995; Erev and Roth 1998; Mookherjee and Sopher 1994; 1997; Chen and Tang 1998).

In their seminal work, Erev and Roth (Roth and Erev 1995) consider three variants of the base model (equations (2.2) and (2.3)); later referred to as ER models). The first uses cutoff parameters for high and low selection probabilities: Actions above the upper cutoff are played with probability 1, below the lower with probability 0. In the second model, a parameter ϵ sets the probability with which a random action is chosen. This allows for persistent experimentation. The third variant includes a recency parameter $\phi, 0 < \phi < 1$, which weights the importance of past payoffs whenever the action strengths are updated: $q_k(t+1) = q_k(t)\phi + \pi(t)$. These models are applied to a large number of games, as the main motivation for RL here is to find a learning model that predicts well over as many classes of games as possible. Erev and Roth use ultimatum games, bargaining (market) games, and simplified best-shot games. Except for ultimatum games, they find that all three RL models predict actual behaviour well, which also happens to converge to equilibrium predictions. In the ultimatum games however, subgame perfect equilibrium (where the first mover demands the greatest possible share for himself) is not reached. Predicted as well as

actual behaviour did not converge to equilibrium. Moreover, experimental data showed differences in medium- and long-term outcomes. The RL model could replicate such switches.

Later, in Erev and Roth (1998), they apply simple RL to a wider collection of experimental data based on mixed-strategy games; this makes convergence more difficult, since no player has an incentive to stick to a pure strategy. Additional to the simple model, they allow for alternatives with more sophisticated learning. Three models are compared: Model (1) is simple RL as in equations (2.2) and (2.3). Model (2) combines forgetting and generalisation, i.e. $q_k(t+1) = (1-\phi)q_k(t) + E_k(j, \pi(t))$, where E is a function determining how playing strategy k affects similar strategies j. In the considered 2-player games, they set $E_k(j, \pi(t)) = \pi(1-\epsilon)$ if $j = k$ and $E_k(j, \pi(t)) = \pi(t)\epsilon/(M-1)$ (where M is the number of pure strategies) otherwise. That is, depending on ϵ, players generalise rewards in a way that leads to experimentation among similar strategies. In model (3) some simple beliefs are integrated in the form of limited (only own payoffs are known) and full information (also opponents' payoffs are known) fictitious play. In the first case, the update function is augmented by an expected payoff parameter, in the latter the action probability is determined considering the value of alternative strategies. After fitting the data, they find that adding more knowledge in the form of beliefs and expectations does not add to the predictive power of RL. Usually, the simplest models predict behaviour accurately. Adding adaptation parameters like recency and experimentation improves the fit of simple RL, but fictitious play does not.

Sarin and Vahid (1999) describe the Payoff Assessment learning model (PA), which uses average payoffs instead of cumulative payoffs, and chooses deterministically the action with the highest expected payoff. Applying it in Sarin and Vahid (2001) to the same data as Erev and Roth did in Erev

and Roth (1998), they find that this model predicts the data at least as well as simple RL.

Mookherjee and Sopher (Mookherjee and Sopher 1994; 1997) conducted experiments with constant sum games. In their early experiment only two choices were available. Players learnt to play their minimax strategies. In Mookherjee and Sopher (1997) they find that experimental results deviate considerably from equilibrium predictions in games with at least four strategies. Instead of cumulative payoffs, here q_k is some average measure of action k. Furthermore, they use the exponential selection function

$$p_k(t+1) = \frac{e^{\lambda q_k}}{\sum e^{\lambda q_j}} \tag{2.4}$$

(where λ is a choice parameter). After comparing also belief-based learning rules, they further conclude that the RL predictions match the reality closest. Using different averaging mechanisms, their data suggest that players' memory is rather short, and that they form expectations about future payoffs.

Chen and Tang (1998) use a cumulative reinforcement function as in equation 2.2 with an exponential selection rule as in equation 2.4. Applying it to public good provision games, they compare its performance in predicting experimental data with fictitious play as well as the equilibrium prediction. They find that the empirical results deviate from the equilibrium prediction, which predicts the data worst. The RL mechanism fits data better than fictitious play.

Arthur (1993) proposes a model similar to the ER type of models. The action strengths q are updated according to equation 2.2. Actions are chosen according to 2.3. However, the sum of probabilities in the denominator is normalised to a pre-chosen constant C. Let e_t be the random unit vector

defined as

$$e_t = \begin{cases} 1, & \text{x is played at t} \\ 0, & \text{x is not played at t,} \end{cases}$$

. The cumulative update function in equation 2.2 can be written as

$$q_k(t+1) = q_k(t) + \pi(t)e_t \tag{2.5}$$

.

Then, let the cumulative payoff until time t be $v_t = \sum_{s<t} \pi_s$. Let $\Delta p(t) = p(t+1) - p(t)$ denote the incremental change in the probability vector e at time t. Because of equations 2.5 and 2.3 one can write $\Delta p(t) = (\pi_t/v_t)(e_t - p(t))$, that is the incremental impact of new experience diminishes over time at a rate of the order of $1/t$. Arthur proposed a model of the form $\Delta p(t) = [\pi(t)/(Ct^\nu + \pi(t))][e_t p(t)]$. In that model, the incremental impact of the current payoffs on the action probabilities decreases over time at a rate of the power of t, which is estimated from data. This is another way of solving the problem of just accumulating experience over time without possibilities to revise choices. Arthur fits the model to single person multi-armed bandit experimental data and finds no systematic differences between simulated and human learning (from Young (1993),pp.11-13).

2.2.2 Experimental Games Using Combined Belief and RL models

Camerer and Ho (1999) argue that there are two fundamental types of learning, experience- and belief-based learning. They propose a more complex approach to experience learning by combining fictitious play with RL. Their experienced-weighted attraction model (EWA), is described by two central equations:

$$N(t) = \rho * N(t-1) + 1 \tag{2.6}$$

and

$$A_i^j(t) = \frac{\phi N(t-1) A_i^j(t-1) + [\delta + (1-\delta) I(s_i^j, s_i(t))] \pi(s_i^j, s_{-i}(t))}{N(t)} \tag{2.7}$$

$N(t)$ denotes the experience weight, and $A_i^j(t)$ the attraction of strategy j for individual i. $s_i(t)$ is i's strategy at time t, and s_{-i} are the strategies of all other players. The function $I(s_i^j, s_i(t))$ is an indicator function and equals 1 if $s_i^j = s_i(t)$, and 0 otherwise. The payoff π is obtained by player i if he chooses s_i^j, given the behaviour of the other players $s_{-i}(t)$. ρ, ϕ, and δ are the parameters of the model. The initial values of $N(t)$ and $A_i^j(t)$ are priors and may be initialised with some experience level the players already have.

For $N(0) = 1$ and $\rho = \delta = 0$, the model reduces to pure cumulative reinforcement learning. For $\delta > 0$, experience collection is expanded to actions not played by observing the other players in the game. If $\rho = \phi$ and $\delta = 1$, the model reduces to weighted fictitious play; for other parameters, the learning represents a mix of RL and fictitious play.

The action selection function has an exponential form and is given by

$$P_i^j(t+1) = \frac{e^{\lambda A_i^j(t)}}{\sum_{k=1}^{m_i} e^{\lambda A_i^k(t)}} \tag{2.8}$$

where the choice parameter λ determines how strongly differences in the attractions translate into choice probabilities, and m_i is the number of possible actions player i can use. If λ is very large, small differences result in a high probability relative to the smaller attractions. If λ is small, differences are ignored until the distance becomes reasonably large.

Camerer and Ho test this model with data from constant-sum games, among them the games from Mookherjee and Sopher (1997), and compare EWA with random, simple RL and belief-based outcomes. The results show that belief-based learning predicts better than EWA in the simpler 4-strategies games, but worse in more complex 6-strategy games. Contrary to Mookherjee and Sopher (1997), they find that belief-based learning converges better than RL learning, which they attribute to differences in the model. For example, Mookherjee and Sopher allowed similar strategies to influence each other, and they used average instead of cumulative reinforcements; both factors favour their RL rule, while in EWA, these aspects are reflected in the belief component.

EWA has been criticised as being too complex and requiring overly many parameters. Therefore, Camerer and Ho developed in Camerer et al (2007) a simplified version of EWA, 'self-tuning EWA', by fixing most of the parameters and only estimating ϕ and δ with dynamic functions. If a player detects a change in opponents' play, ϕ is adjusted to allowing more experimentation, and vice versa (becoming pure RL in stationary environments). The attention function sets δ to 1 if the foregone payoffs are higher than the actual received payoff, so that alternative strategies are reinforced, and the agent eventually may switch to one of the superior actions. If there is no better choice available, δ is set to 0, thereby supporting an RL-like lock-in into the best response strategy. Comparing the predictive power of full and simple EWA, they find that self-tuning EWA is not as good as the original approach, but produces very similar results. This applies especially if parameters are estimated for the same class of games. Self-tuning EWA predicts better if parameters are estimated jointly for different games.

Chen and Khoroshilov (2003) compare different learning models - EWA, the PA model and simple ER learning - in coordination and cost-sharing

games with two players. They find that PA fits best to the data, followed by EWA and RL. When estimating parameters over different games (pooling) PA does best. An exception is cost-sharing games with an average-cost distribution among players. Under this mechanism, the cost is distributed evenly, and thus experimentation in one agent triggers experimentation in the other players. None of the models converged to the observed data.

Stahl (2000) develops a model in which players learn to choose among different strategies following simple decision rules. Players know the strategies played by their opponents. Analogously to other learning models, rules in the rule space that were successful in the past are more likely to be selected. The rule space can be thought of as composed of basic, or 'archetypical', rules, from which more complex behaviour can be constructed. The evidence of every rule is assessed, and the probability of choosing that rule is derived using an exponential selection rule. This evidence is, e.g., the expected payoff given the opponent's strategy in t-1. Based on such reasoning, Stahl defines five strategies (e.g. strictly dominated vs. Nash equilibrium strategies) which are first tested in experiments, and then fitted to the data. He finds that the model fits the data better than the equilibrium prediction and random outcomes. The model uses nine parameters, which is found to be the required minimum to fit the data well. Furthermore, evidence from the experiments suggests that real humans do not gather evidence about all rules as proposed by the model, but rather focus on subsets.

2.2.3 Analytical Approaches with Simple RL Models

Many authors have analysed the properties of learning rules, and try to establish conditions under which the actions of players converge to the optimal action (in single-player decision problems) or equilibrium (in games). Typically, the proofs for convergence rely on stochastic approximation the-

ory. Early work mostly established results for limited classes of games or simple one-player decisions. Only more recent articles (e.g. Beggs 2005; Hopkins and Posch 2005; Gotts et al 2007) could state more general results for the boundary behaviour for the process, and larger classes of games.

ER models Some authors have analysed the ER learning rule (Rustichini 1999; Laslier and Walliser 2005; Beggs 2005; Rustichini 1999; Hopkins and Posch 2005) in single decision and game contexts.

Rustichini (1999) considers optimal properties of selection rules under full and partial information in a single player context. Under full information the player knows opponents' strategies, under partial information only its own actions. He finds that with a linear rule (as in equation (2.3)), convergence to the optimal choice is guaranteed. It is not with the exponential rule, which weights differences between payoffs higher and thus might speed learning up. Moreover, exponential procedures (as in equation (2.4)) are best in the full information case, but not for partial information: Linear learning is too slow in full information environments, so the process is more likely to lock into sub-optimal interior points of the strategy space, rather than the optimum.

According to Laslier et al (2001) the cumulative RL problem can be seen as an urn model, from which balls are selected with unequal probability over the repetitions of the game. Describing this process with ordinary differential equations (ODE), they first analyse the resulting stochastic process for single player situations and show that the process converges to choosing only payoff maximising actions. For 2x2 games they state that the ER rule converges with positive probability to a Nash equilibrium. If the game has two pure equilibria, the process converges with positive probability to any one of them, but not to a mixed equilibrium. However, they cannot prove

that the process converges with probability 1.

Building on stochastic approximation theory, Beggs (2005) considers 2x2 constant-sum games with unique pure or mixed equilibria and generalises Laslier et al (2001). Players using RL cannot be forced permanently below their minimax payoff, independent of their opponent's strategy. Similarly, dominated strategies are always eliminated over the course of time. If both players play RL, the probability that both players converge to the unique equilibrium, tends towards 1.

Hopkins and Posch (2005) provide more general results about the relationship of the RL processes with the well-analysed replicator dynamics approach from evolutionary game theory (Smith 1982). They find that Arthur's model (Arthur 1993) as well as ER-type models converge only to boundary points which are a Nash equilibrium. This is easier to show for the Arthur model because the action strength updates (step sizes) are of the same size, while the reinforcements in ER can change at different rates. They show that RL will not converge to boundary points that are linearly unstable under the replicator dynamics.

Averaging models In PA, a decision maker faces for a number of times an identical decision problem. The players assess expected payoffs myopically by estimating the expected payoff using average returns per actions. They choose the action with the expected maximum payoff (i.e. choice is deterministic). Sarin and Vahid (1999) show that this model converges to choosing the objective maximin strategy if learning is slow. If players are more likely to experiment, players converge to the strategy yielding the maximum possible payoff.

Aspiration level models The reinforcement problem in aspiration level models has been also been studied by several authors, and has been surveyed in-depth by Bendor et al (2001a). Here, some representatives of this approach are described.

Gilboa and Schmeidler (1996) present a case-based reasoning (CBR) approach. The decision maker faces a number of different situations or 'states', and must make a choice in such situations. In dynamic environments, aspiration level (AL) updating rules have to be ambitious enough to search for the best result in various situations. In more static environments, it must be realistic, i.e. close to actual payoffs. Both properties must be combined, as a way to search ambitiously for a best strategy, and then to stick to this choice after the expected values of the strategies can be estimated. They show that under these conditions, a case-based decision-maker can learn to become an expected-utility maximiser.

Extending their work on RL with fixed AL, Boergers and Sarin (1997) develop a model with endogenous aspirations and cumulative rewards. In Boergers and Sarin (2000), a single player chooses between two strategies. They show that the process can converge to the optimal choice. Endogenous aspiration levels improve performance by avoiding high dissatisfaction with even the best available strategies, but can lead to probability matching. During probability matching, both strategies are played at the same probability at which they generate benefits, whereas optimal strategies should be played with probabilities close to 1 for behaviour to be considered 'rational'. This can happen when the initial aspiration levels are too high, so that also dynamic adaptation of the aspiration level cannot lead to a lock-in.

While Boergers and Sarin and Gilboa and Schmeidler establish results for single player decision problems, other authors extend the results to

games. Karandikar et al (1998) first analysed a prisoner's dilemma. The aspiration levels of both players are updated simultaneously with the received reward, and approximate long-run averages. The main result is that cooperation is sustained if there are no trembles (i.e., externally imposed changes or noise on the AL's) to the AL's and the speed of updating the AL's is low. Introducing perturbations into the AL changes the process, and may lead to different equilibria. However, in the long run, the process returns to the cooperation path. The intuition behind these results is that the mutual dissatisfaction with non-cooperative payoffs triggers experimentation until some state is achieved that yields high enough satisfaction (the point where AL and current payoff converge).

Karandikar et al (1998) is modified and extended to arbitrary games and a larger class of learning rules in Bendor et al (2001b). Similarly, Napel (2003) applies the model to an ultimatum game and shows that in the long run players almost surely achieve the equilibrium state. Which equilibrium depends on the initial conditions and the stability of aspirations, which are allowed to vary randomly. If such trembles are rare and learning is slow, the available surplus will be shared efficiently. If there are perturbations in the aspiration level, any equilibrium is supported.

Gotts et al (2007) look at the behaviour of the BM rule with aspirations in a prisoner's dilemma, generalising earlier insights of Flache and Macy (2002). They show that the system has two attractors - either a mixed strategy equilibrium (a so-called self-correcting equilibrium SCE) or both players cooperate with probability 1. If learning is slow, the system converges in the long run to cooperation. In the medium run however, the process moves towards the SCE. RL thus can exhibit very different results depending on the length of the period considered.

2.2.4 Cognitive Approaches

This section reviews two approaches of a more cognitive nature stemming from Artifical Intelligence AI. Still being based on own experience, they provide mechanisms to make the agent aware of different conditions in the environment.

CLARION The cognitive architecture CLARION (e.g. Sun and Slusarz 2005) was designed to capture implicit and explicit learning processes in humans. The main assumption is that there are two different levels of learning: A subsymbolic 'bottom' level and a symbolic 'top' level. The 'bottom' level represents low-skill, often repetitive tasks for which learning proceeds in a trial-and-error fashion. Knowledge on this level is typically not accessible, and it is difficult to express such skills with language. On the symbolic level, knowledge is directly accessible and can be expressed with language. This level typically represents more complex knowledge. It can be acquired by experience, but also by explicit teaching.

The input state is made up of a number of dimensions, and each dimension may specify a number of possible value or value ranges. Action selection takes place using RL in the bottom level, or by firing production rules on the top level. Which level is used is determined stochastically. After the action was performed, top and bottom levels are updated with the feedback received from the environment.

At the bottom level, the RL mechanism is implemented with a neural net. The input layer is constituted of the values of the input state. Three intermediate layers are used to compute Q-values (allowing memory of action sequences), while the fourth layer chooses an action according to standard reinforcement learning (similar to equation (2.10)).

At the top level, the rule conditions are constructed out of the input dimensions, their consequents from actions available to the agent. The rules are, for compliance with the bottom level, implemented as network. Rule extraction, specialisation and generalisation are determined by feedback from the subsymbolic level: If there is no rule matching the current state and the action performed well according to some performance criterion, a new rule is created with the current state as the condition, and the performed bottom level action as consequent. If rules matching the current condition exist and the action was successful, the matching rules are replaced by a generalised version by adding another input element to the condition. The covered rules are deactivated, but might become reactivated if specialisation is applied to the new rule at a later stage. Conversely, specialisation means the removal of an input value from the condition and is triggered when the result of an action was not successful in the specified condition. Deactivated rules are reactivated if the specialised rule does not cover them any more. An information gain measure that estimates the performance of rules under different conditions serves as the success criterion.

This model is applied in Sun and Naveh (2007) to a 'stone-age economics' simulation in which agents belonging to a group collect and contribute food. An agent might cheat and not contribute, which is punished with some probability. They show that their adaptive agents are able to reproduce results of the same model with more deterministic strategies investigated before (Cecconi and Parisi 1998). They also investigate the properties of the emerged survival strategies. For example, it turns out that relying more strongly on the top level enhances performance, and that higher probabilities of rule generalisation are beneficial only when less importance rests with the bottom layer.

Learning Classifier Systems Learning Classifier Systems (LCS) also aim at the extraction of rules. The basic idea is to start with a set of initial rules (classifiers) and to evolve this set over time by application of mechanisms for modification, deletion and addition of new rules. Whereas earlier LCS, as introduced by Holland (1975), relied mostly on the Genetic Algorithms paradigm, newer versions have more in common with RL approaches and so have also been described as generalised RL (Sigaud and Wilson 2007).

An LCS consists of a population of classifiers. A classifier contains a condition part, an action part, and an estimation of the expected reward. Typically, the condition part consists of the three basic tests 0 (property does not exist), 1 (property exists) and #. # represents a generalisation and stands for both 0 or 1. A classifier has one action as a consequent, but typically several classifiers match a condition in the environment and hence compete with each other. The action to be executed is then selected according to some RL mechanism (e.g., the ϵ-greedy policy, which selects the best-performing action at a rate of ϵ, $0 < \epsilon < 1$ tries a random action).

Many LCS use a Genetic Algorithm to create new rules by selecting and recombining the fittest classifiers from the population (where fitness is, e.g., the expected reward received from the environment). A covering operator is called whenever the set of matching classifiers is empty. The operator adds a classifier matching the current situation with a randomly chosen action to the population. Sophisticated systems may limit the population size, and add corresponding eviction and generalisation procedures.

Newer families of classifier systems, like anticipation-based classifier systems (ACS, Butz (e.g 2002)), do not rely on evolutionary methods. They extend the classifier representation with the description of the next state and

build a model of transitions. A specialisation mechanism is applied when the classifier oscillates between correct and incorrect predictions, indicating that a splitting of the condition might improve the match. Generalisation is based on complex algorithms that estimate whether generalisation will result in an improvement (see also Sigaud and Wilson (2007) for an overview of LCS).

Applications in Economics have usually used Holland-type classifiers. Markets of different kinds have been modelled using LCS, for example, the market for electricity (Bagnall and Smith 2005), for fish (Kirman and Vriend 2001b), or stock markets (e.g. LeBaron et al 1999).

In Bagnall and Smith (2005), the UK electricity market is modelled. In the model, there are a number of electricity generating agents. Each agent must produce an offer bid per day for the amount of electricity it wants to produce. The strategies are determined by three factors - capacity constraints, demand and capacity premiums (for particular time slots in a trading period). By this, a 10-bit vector of states, denoting different demand, constraint and premium situations is constructed. The model is used to model various scenarios. For example, they reproduce actual, observed bidding behaviour.

Kirman and Vriend (2001b)'s model represents a wholesale fish market, in which buyers and sellers are matched. Buyers resell the fish, and their payoff is given by the difference of the prices they pay and a fixed price they receive. Analogously, sellers' profit is determined by the difference of their costs and the selling price. Classifiers are used for several decisions, such as deciding stock levels, or buying and selling prices. Furthermore, buyers may become loyal by choosing to return to a seller; sellers remember their customers and may reward loyalty by lowering their ask price. It turns

out that loyalty develops as buyers and sellers realise simultaneously the benefits: Returning customers allow better planning of a seller's stock and continuous profit flow, for which lower prices are accepted; because of these, customers learn to return.

The stock market model of LeBaron et al (1999) aims to reproduce actual stock market behaviour in an artificial stock market. In the market, there are trader agents whose task is to make forecasts about the future price of assets. The expected price is used in their demand functions, which then determines the amount of assets to purchase. The agents base their forecasts on hypotheses or candidate rules, of which a single agent maintains 100. These rules map conditions of the environment into forecasts. The state vector is 12 bits long. The conditions are given by dividend/price ratios and comparisons between current price and average prices, which describe the value of an asset given the market conditions. LeBaron et al (1999) are able to reproduce features of price time series taken from real markets.

Summarising, LCS are a way to represent learning where the environment is dynamic and unclear which possible rules are best for the agent's performance. They are, in principle, a directed search among candidate rules: Starting from a large set of possible rules, those are selected that perform best in the environment the agent is in. Weaknesses of LCS have been handled in the newer approaches - for example, by modelling state transitions. However, the mechanisms when to apply generalisation and specialisation are complex. In this sense, LCS can become relatively 'heavy' models of mental processes. It has been suggested that using simpler RL methods is sometimes easier and better tractable (e.g. Holland et al 2000).

2.3 Concept

As the literature overview in the previous section showed, there is substantial literature, mainly in the area of simple games. Fewer authors attempted to develop cognitive strategic models. Each approach has its limitations with respect so ACE modelling. Thus, a cognitive architecture covers psychological details social scientists are often not interested in. LCS are a rather technical approach to learning. For some domains and problems, the representation system might not be adequate (Schuurmans and Schaeffer 1989). In particular, the representation of knowledge as bit strings may introduce problems. For example, it is difficult to represent more abstract knowledge like relational operators such as greater, smaller etc. To cover large value spaces, it would be necessary to represent each single value as bit in the string. Thus, representing fish prices from 0 to 1000 in Kirman and Vriend (2001b) would become difficult, or at least require implicit knowledge about the domain to set up the classifiers adequately.

The main contribution of the computational approach presented here is the formulation of a learning model that covers simple as well as more cognitive modes of learning. From a theoretical point of view, it should be a mixed model. As a computer model, it should be valid in the sense of reflecting simple, but realistic decision making, and simple in the sense that it focuses only on decision mechanisms in social and economic contexts. It should therefore be more specific as a cognitive architecture, and have more natural and broader representation features as LCS.

A simple framework covering these goals is readily available since the early contributions to bounded rationality (Simon 1956a;b) and actually has not changed substantially since then. This framework is based on the following components:

- The set of behaviour alternatives A
- The set of choice alternatives A' for bounded rational or computationally less powerful individuals; this set may be only a subset of A.
- Possible future states S
- Payoffs connected with S, represented as a function of S, $V(s)$.
- Probabilities for S. There is uncertainty which state occurs after a particular behaviour, i.e. there may be more than one.

Bounded rational individuals do not typically know the mapping from behaviour alternatives A to future welfare $V(s)$. A possible strategy to learn about the occurrence and the desirability of these future states is according to Simon: Start with a mapping of each action alternative $a \in A$ to the whole set of S. Using a utility function such as $V(s) \in \{-1, 0, +1\}$, find $S' \subset S$ such that (expected) $V(s) = 1$. Then gather information to refine the mapping $A \rightarrow S'$ (i.e., which actions lead to which result under certain conditions) and search for feasible actions $A' \in A$ that map to S' (Simon 1956b). In other words, an agent's goal is to find the states which satisfy its needs, by exploring the state-action space by applying alternative behaviours.

The translation of Simon's framework into an executable algorithm can be captured best with the concept of mental models. A mental model is an internal representation of an external reality. The agent builds it using experience, its perception, and its problem-solving strategies. A mental model contains minimal information, is unstable and subject to change and used to take decisions in novel circumstances. A mental model must be 'runnable' and able to provide feedback on the results. Humans must be

able to evaluate the results of actions or the consequences of a change of state (Markham 1999). It is assumed that an agent is only interested in its own welfare, and its goal is to find suitable behaviour strategies that optimise utility under different conditions. Information processing and memory are costly, so that the internal model being built has to be minimal and efficient with respect to the agent's welfare. The main principles an algorithm has to account for can roughly be summarised as follows:

Evaluating cognitive cues In any state of the environment, the agent must be able to choose an action. If low or even negative rewards are experienced, the agent can attempt to apply a different action. If this fails to improve the agent's welfare, this is a hint to pay attention to more cues from the environment and distinguish better between situations.

Deciding what to know Paying attention to all cues is computationally expensive and memory limited; humans must filter out certain aspects of their perception in order to decide and act effectively. The agent has to 'decide what to know' (Rubinstein 1998). What information is useful depends on how it helps to improve the agent's welfare. This can only be tested by using the accessible information while acting. Since the usefulness is unknown initially, the decision procedure can be seen as a search over all possible state-action mappings. If the agent is satisfied with a mental model containing a subset of these mappings, it might stop searching for a better model or decrease its search intensity. As a rule of thumb the agent follows the most promising direction. If a certain configuration of mappings increases welfare, it tries to improve this configuration, e.g. by specialising the contained information.

Updating a cognitive model If the environment changes, some aspects of the internal model might become obsolete. The agent will then experience a change in utility. In certain states, learning a new behaviour might be sufficient. However, it might also be that the representation of the state is not accurate anymore (e.g. a new type of agent appears). In this case, the representation has to be changed, e.g. by removing old representations and start the search process anew for certain parts of the model.

A similar idea has been used in Gifford (2005). In this model, agents have limited information about future outcomes of opportunities (e.g. stock returns), and have to decide whether to evaluate new, or to stick to old behaviours (e.g. buying a new stock). Attention is a scarce resource, so that evaluating alternatives becomes costly. It turns out that the higher this cost is, the more 'irrational' the behaviour; if cost is neglected, and agents can spend more effort on evaluating future expected states, behaviour approximates more rational decision-making.

2.4 The Algorithm

The basic idea of the 'Bounded Rationality Algorithm' (BRA) is to build an internal, flexible model of the environment the agent lives in. The environment is accessible by the input state s defining the current 'situation' the agent is in. The input state is matched with an internal symbolic representation $C_i \in C = \{C_1 \dots C_n\}$ of the state. The agent then chooses an action according to the general form $r_i : C_i \rightarrow A$. A is the action set, C is the set of all possible conditions that can be generated from the input dimensions, and C_i is a collection of conditions derived from C.

The next paragraphs develop the algorithm in detail.

2.4.1 Reinforcement Learning

RL is used to implement the dynamic aspect of knowledge generation in the model. In each state agents learn by trial and error which action to apply in a given state. Successful actions are rewarded. Actions which yield a higher reward are selected with a high probability in the future, whereas bad actions, receiving a lower reward, are selected less often. The history of these reinforcements is summarised as action strength q. Whenever an action a has been applied, the strength is updated with the reward $p(t)$ observed for that action by the following equation (Sutton and Barto 1998):

$$q(a_t) = q(a_{t-1}) + \gamma(p(t) - q(a_{t-1})) \tag{2.9}$$

This action-value function updates the strength of the current action based on the weight γ of previous experiences and the current reward. It is a method to approximate the true value of $q(a)$ out of a sample of values. The smaller γ, the stronger the impact of past experiences; conversely, for $\gamma = 1$ only the reward of the last action is considered, and all previous experiences discarded. Thus γ determines the speed of updates.

In the next step, the action probability is calculated according to the selection function:

$$pr(a_{i,t+1}) = \frac{e^{q(a_i)/\alpha}}{\sum_j e^{q(a_j)/\alpha}} \tag{2.10}$$

This exponential selection function determines each action's selection probability depending on its own strength relative to the strengths of the alternative actions. The parameter α is a parameter that determines the rate of exploration. The influence of the action strength on the selection probability decreases as α grows. For large α, the selection probabilities approach uniform values. Sutton and Barto (1998) report that for many problems, α values of about 0.1 turned out to achieve a good balance between exploration and exploitation of learnt behaviour. For many problems, α values

approaching 1 translate into selection probabilities smaller than the original action strengths, so that too large values quickly stop being useful for the learner. Finally, as $\alpha \to \infty$, each choice becomes equally likely.

2.4.2 State Space Partitioning

Learning by doing as described above happens for a given state s. This section describes how states are represented and perceived in the agent's internal world model.

Representation The state s is represented internally as a collection of attributes $\{att_1 \dots att_i\}$. Each attribute can have a number of possible values, for example nominal values such as 'low' or 'high', or numerical ranges, e.g. 0-1000. Attributes are connected by simple predicate logic. For example the predicate '(profit=low *or* profit=medium *or* profit=high) *and* (sales $0 <$ sales < 1000)' could describe the situation of a firm in the dimensions profit and sales. This representation is called a 'state descriptor', and formally denoted C_i. To each state descriptor actions are bound from which the action policy for this state can be learnt. In the firm example, actions could be an array of price levels. This binding constitutes formally the mapping $r_i : C_i \to A$.

The agent starts with a model covering all possible states. This initial model contains a root state description or a set of disjunct root state descriptions; each root descriptor contains all attributes with their value spaces relevant for this partition, thus the coarsest representation possible. In consecutive time steps, specialisations are developed stepwise by the application of a heuristic search method. For this, the space of state descriptions is represented as a tree, where nodes at higher levels contain coarser, and nodes at a deeper level of the tree finer mappings. Finer grained descrip-

tions are 'expanded' from the predicates at higher levels. Coarser grained descriptors can be generalised again if the more detailed descriptors do not perform better than the parent. Which descriptions are expanded depends on a heuristic evaluation function, which here is the agent's utility. Each state descriptor has a value that describes this utility. The task of the search process is thus to find the level of detail that describes the environment in such a way that generates the highest welfare for the agent.

Depth-first search principle The path the expansion mechanism takes follows a depth-first search paradigm. If finer grained descriptions increase welfare this path is followed further, that is, the mechanism assumes that the most accurate state descriptions are best. Using a tree-search approach, this corresponds to a process in which a single node on level h is expanded to level $h + 1$ according to some performance criterion, while the siblings on level h are not taken into account. The path this process takes is represented by the 'search path'. Each node the process expands is added to this path, and removed when it is generalised. The search path is thus a list which contains all nodes of the tree that are relevant for the model specialisation and generalisation methods. These methods are described in the next paragraphs.

State expansion mechanism Before the internal model is updated, the agent acts in its environment over a period μ. During this period, the value of existing state descriptions $R = \{r_1 \dots r_n\}$ is updated using feedback from the environment. After each μ steps, the state expansion mechanism is applied: First the node r_{expand} with the highest value on the search path is selected. If the search path is empty, a root node is selected. From there, the next level of the tree is expanded by partitioning the value spaces of the attributes constituting the conditions of r_{expand}. For attributes having

discrete values, one value is picked randomly. Attribute values representing numeric ranges are split in half. For each partitioned attribute a new condition is created containing the partitioned attribute values or value range, and the remaining original attribute values (i.e. the number of successor nodes equals the number of attributes $\times$ 2 in the original condition). The conjunction of the predicates of the resulting level (after reduction) is equivalent to the expression of the parent node. By mapping A to each newly created condition set the new descriptors R' are generated. The path from each $r' \in R'$ up to the root node is set as search path (without duplicates). The conjunction of state descriptions with no children in the tree is then equivalent to the initial state description. The RL mechanism selects actions only from the matching leaf descriptors. There might be, depending on the paths that have been expanded, overlapping descriptors. In this case, for deciding which state is activated some conflict resolution has to be applied. This could be the selection of random node, or the node with the highest value. In the implementation used for the models of the thesis (see also appendix A.2.3), a random node is selected.

For example, going back to the firm example above, of the initial, exhaustive description $C'_{initial}$ = (profit=low *or* profit=medium *or* profit=high) *and* (0 < sales < 1000) the attribute profit is selected, and of its value range the value 'high'. The value space of the attribute is divided into the expression 'profit=low or profit=medium' and 'profit=high', respectively. The resulting specialised state descriptions are C'_1 = (profit=low *or* profit=medium) *and* (0 < sales < 1000) and C'_2= (profit=high) *and* (0 <sales < 1000). Analogously, the sales attribute is split in two intervals and two successor descriptors generated, so that four successor descriptors are created. Figure 2.1 depicts how a search path is generated by this process.

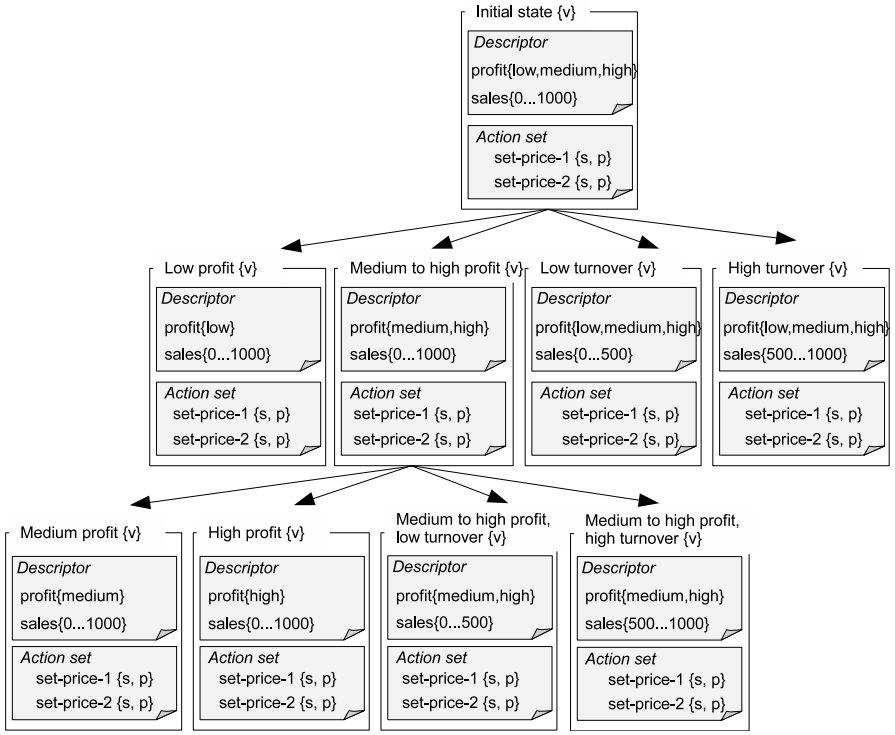


Figure 2.1: The agent's representation of the state space after partitioning all possible profit situations. Each state is described by a set of attributes and an action set. Actions executed in this state are updated with strengths s and selected with probabilities p, which are determined by rewards. The rewards also determine the state value v.

Model specialisation and generalisation With the state expansion mechanism it is possible to specialise the conditions in the state-action space in many ways. A heuristic evaluation function determines the direction of this process. This function is calculated as follows: First, the value of a state at time t is calculated as

$$v(r,t) = v(r,t-1) + \lambda(q(a_t) - v(r,t-1)) \tag{2.11}$$

where $q(a_t)$ is the reward of the executed action in the state described by r. The function approximates an average of the state description value; the

speed of update is governed by the parameter λ *.

Before an expansion happens, some constraints have to be satisfied: The parameter χ limits the maximum number of nodes the tree can have, i.e. the maximum number of situations the agent can differentiate. New states can only be evolved at the cost of 'forgetting' other state descriptions (see below for deletion). Furthermore, since the deletion of nodes might occur, it is possible that state descriptions that were deleted are expanded again, so that endless cycles of generalisation and specialisation occur. The right balance has to be found depending on the stability of the environment; preventing many visits of identical descriptions too early can be harmful if the environment changes; on the other hand, the agent should be allowed not to become trapped into useless expansion/retraction cycles. So to speak, the agent is taught that constantly trying the same without effect is worthless. To tune this balance, a function with a cost parameter $\zeta, 0 < \zeta \leq 1$ is used to compute a value determining whether the successor description should be developed or not: The better a state descriptor compared with the average performance (measured by the average reward at time t, $g_t = \kappa(r(a_{i,t}) - g_{t-1})$ †) and the smaller ζ, the more frequent (recurrent) expansions beginning from that state descriptor are allowed (equation (2.12)).

$$expand(r) = \begin{cases} true, & \text{if } expansions(r) = 0 \text{ or} \\ & \zeta \times expansions(r) \times g < v(r,t) \\ false, & \text{otherwise} \end{cases} \tag{2.12}$$

A state description might lead to a good solution strategy, but if only

*In the implementation used for the models of this thesis, λ is fixed at 0.5. Since v represents a part of the environment, updates should be not too fast. The medium value has been chosen as the norm; reasons for adjusting this value in simulations might be given, but did not arise in this thesis.

†Here again, the update speed parameter κ was set to 0.5 for the simulations in the thesis. Since g is supposed to be a representative value of reasonably large sample, the average value of the possible interval $0 \ldots 1$ has been chosen.

rarely visited is of limited value (they only use up scarce memory space and processing capabilities). Therefore, a heuristic function h used by the process is the state-value weighted by the number of its activations to account for the recency of the value:

$$h(r,t) = v(r,t)\frac{activations(r)}{t} \tag{2.13}$$

The search process selects the node with the maximal heuristic $h(r,t)$ in the search path, if the *expand* condition is satisfied. In accordance with depth-first principle described above, the expandable set of nodes in the search path are the leaf nodes. $h(\cdot)$ is only applied to those nodes.

Before new states are developed after μ steps, the state descriptions of the current level of the tree h may be deleted if they did not outperform the value of their parent states (performance could be, e.g., the average of the state description values). This is called rule generalisation. A rule generalisation is the reversal of a finer grained state back to its original parent state. Generalisations can thus only take place if at least one expansion has taken place, as the initial state is the all-encompassing state. Analogously to rule specialisation, the generalisation process sets in after a certain time ν. While ν is a parameter, the difference between ν and μ should be reasonably large to allow some re-sampling the state values $v(r,t)$ of the parent node in case of a contraction. By this fine tuning feature, the algorithm can correct a wrong search direction before deciding on the next expansion at the higher level $h-1$. If the $|\nu - \mu|$ is too small, cycles are more likely: Since the parent node has had the largest value in the past, the same 'wrong' expansion will be made again if there are to few updates, which possibly decrease the value to their current true value.

Figure 2.2 illustrates how the initial state is expanded and which states are matched against s. For clarity, predicates are only indicated.

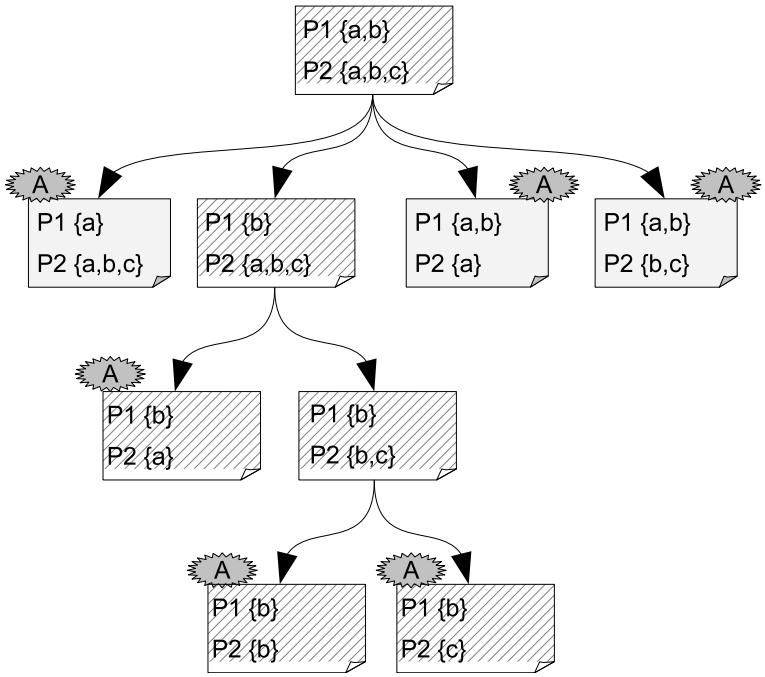


Figure 2.2: Representation of the agent's search space at a particular time. Leaf nodes are active nodes which are matched against s. The hashed nodes represent the search path along which generalisation and specialisation takes place. P... represent the predicates describing the state.

As an example of the specialisation and generalisation process, figure 2.3 shows a possible path of expansion and retraction of nodes. For clarity, only the values of the nodes are depicted.

Avoiding local search optima The search process proceeds in a certain direction. On its way down to more specialised descriptions, it becomes difficult to revert it. Since the environment is dynamic the search path may become suboptimal. There is no back-propagation of values, e.g. an update of the successor states with a discounted value of the current state, so that more general descriptions higher up in the tree or in other partially developed paths can have higher, although outdated values. To leave a certain path and develop different directions in the tree might be difficult; in the worst case, the current deepest level might decrease in value, become

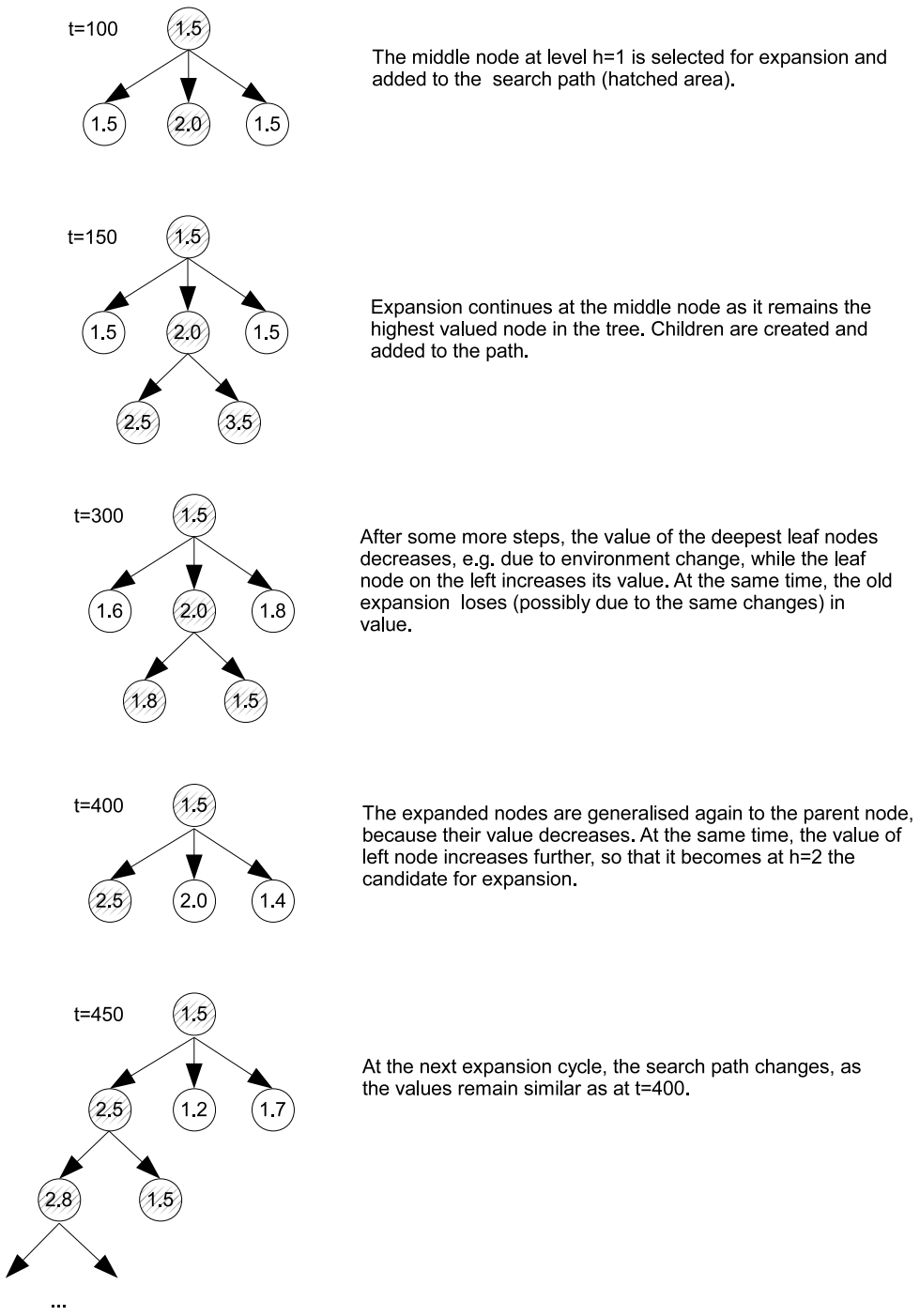


Figure 2.3: Example of the agent's search space at particular time steps and when nodes are specialised and generalised. The hatched nodes represent the search path, the numbers describe the value of the state descriptions.

deleted and developed again so that a circle develops. To get back to a better expansion node can take a long time or even be almost impossible. To prevent such situations, it is possible to switch the search path. Although node values higher up in the tree might no longer be up-to date, the agent uses these values as a hypothesis that they are more promising than the current path. Switching happens with probability $\rho, 0 \leq \rho \leq 1$, in which case the highest overall value in the tree is selected as the new expansion point. The path from the root to this node becomes thereby the search path.

Figure 2.4 illustrates the switching process. It shows that the result is similar to generalisation and specialisation. The difference is that the new path was not reachable because the deepest leaf nodes have a higher value than their (unchanged) parent. With the switching procedure, there is a chance that this trap is left.

The complete algorithm is summarised in pseudo-code in the next section.

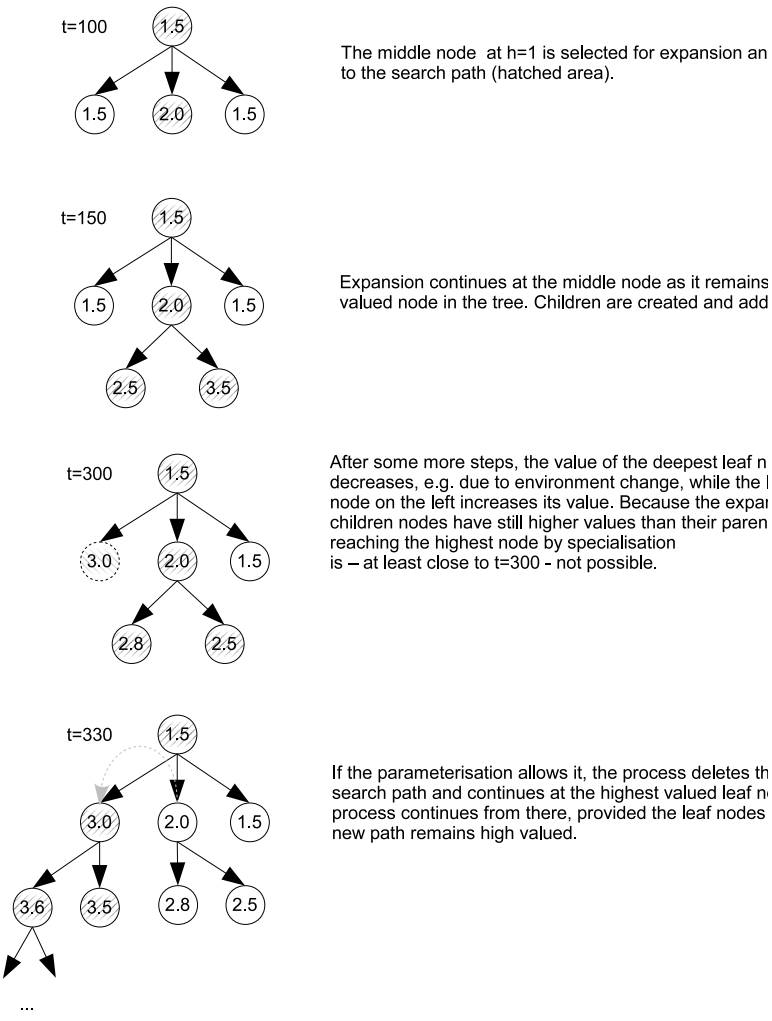


Figure 2.4: Illustration of how BRA avoid local search optima. The hatched nodes represent the search path, the numbers describe the value of the state descriptions.

2.4.3 The Complete Algorithm

This section summarises the algorithm in pseudo-code.

Table 2.1: Summary of notation

Name	Description	Value range
γ	discount parameter for reward	$0 \ldots 1$
ν	interval at which underperforming rules can be deleted	$0 \ldots \mu$
μ	interval at which new rules can generated	$0 \ldots \infty$
ζ	cost parameter determining the frequency of re-exploring already visited paths	$0 \ldots 1$
χ	maximum number of nodes	$1 \ldots \infty$
ρ	probability for switching the current search path	$0 \ldots 1$
p	payoff (reward)	$0 \ldots \infty$
g_t	average payoff (reward) until time t	$0 \ldots \infty$
A	action set of actions a	
q_{a_t}	strength of action $a \in A$	$0 \ldots \infty$
pr_{a_i}	action selection probability of action a_i	$0 \ldots 1$
C_i	conditions that can be generated from input dimensions S	
r_i	A state-action mapping $C_i \rightarrow A$	
$v(r,t)$	The state value function	
$h(r,t) = f(v(r,t))$	The heuristic selection function	

{Setup and Initialisation}

Define the time discount for action updates γ
Define the update-cycle μ
Define the delete-cycle $\nu, \nu < \mu$
Define the cost of expansion ζ
Define the maximum number of states descriptions χ
Define the probability of switching the search path ρ
Define the search path $search\_path$ as a subset of R
Define $expansions(r)$ as a function counting the number of expansions from r
Define $activations(r)$ as a function counting the times r matched a state

Define $parent(n)$ as the parent of a node n in the state-tree $T(R)$

Define $children(n)$ as a function returning all children of a node n in $T(R)$
Define $uniform(x \dots y)$ as a uniform random distribution in the interval $x \dots y$

$q(a) = 0, \forall a \in A$
$C_1 \leftarrow\!\shortmid S$
$search\_path \leftarrow r_1 : \{C_1 \rightarrow A\}$

repeat

{Reinforcement learning}

observe reward $p(t-1)$ received after executing a_{t-1}

$g_t = g_{t-1} + \frac{1}{2}(p(t) - g_{t-1})$
$q(a_t) \leftarrow q(a_{t-1}) + \gamma(p(t) - q(a_{t-1}))$

$v(r,t) \leftarrow v(r,t-1) + \frac{1}{2}(q(a_t) - v(r,t-1))$

$activations(r) \leftarrow activations(r) + 1$

compute situation $s \leftarrow\!\shortmid S$

find the most specific mapping $r_a \in search\_path$ matching s

$pr_{a_{i,t+1}} \leftarrow \frac{e^{q_{a_i} * \alpha}}{\sum_{j,j \neq i} e^{q_{a_j} * \alpha}}, \forall a \in A_{r_a}$

select action a_t from the resulting distribution and execute a_t

{State space partitioning}

{Expand}
if rest$(\frac{t}{\mu}) = 0$ and $|R| < \chi$ **then**
  $r_{expand} \leftarrow \max h(r,t), \forall r \in search\_path$
  if $\zeta \times g_t \times expansions(r_{expand}) < v(r_{expand}))$ **then**
    partition r_{expand} according to expansion mechanism into $R' \leftarrow \{r'_0 \dots r'_n\}$
    initialise the value of the new states with $v(r_{expand,t})$
    append R' as children of r_{expand}
    add R' to $search\_path$, remove siblings of R
    $expansions(r_{expand}) \leftarrow expansions(r_{expand}) + 1$

end if
end if

{Delete obsolete mappings}
if rest$(\frac{t}{\nu}) = 0$ **then**
{determine the most recent expanded mapping $r_{expanded}$ and its children CH}
$CH \leftarrow \{ch_1, \ldots ch_n\} \subset search\_path$, $children(ch_i) = \emptyset$
$r_{expanded} \leftarrow parent(ch_i)$
if $v(r_{expanded}, t) > \frac{1}{|CH|} \sum_{i=0}^{|CH|} v(ch_i)$ **then**
delete CH
end if
end if

{Avoid local search optima}
if $uniform(0, 1) > \rho$ **then**
clear $search\_path$
$r_{max} \leftarrow \max v(r, t), \forall r \in R$, $children(r = \emptyset)$
add the path from r_1 to r_{max} to $search\_path$
end if

until end of simulation

2.4.4 Compact Notation

After the various mechanisms have been described in detail, the following conventions might be useful in describing the system in a more concise way:

An agent's state of mind is represented by a set of state-action mappings R. There can be k distinct sets of state-action-mappings. Each state-action mapping $R^k \subseteq R$ consists of a symbolic representation of the state, denoted by C^k. C is a simple propositional system $\mathcal{L}$ of formulae Z and logical operators Ω, $\mathcal{L}^k = \mathcal{L}(Z, \Omega)$, where a formula consists of terms (variables and constants) and relational operators. The operation succ($\mathcal{L}^k$) partitions the formulae in $\mathcal{L}^k$ into m subsets $\mathcal{L}^k(1 \ldots m)$. By successive application of *succ*, i new successors C^k can be generated, labelled C_i^k. The corresponding $\mathcal{L}^k$ is

augmented by the index i to identify it uniquely: $C_i^k := \mathcal{L}_i^k$. Denoting with l the number of *succ* operations applied from the initial representation, from each $C_{i,l=0}^k$ new symbolic representations can be generated until $\text{succ}(\mathcal{L}_{i,l}^k) = \emptyset$. The action set remains constant per k.

Definition 4. *A complete state-action-mapping during the process of state-space partitioning can shortly be described with* $R_{i,l}^k : C_{i,l}^k \rightarrow A^k$. *R denotes the set of mappings, C the set of symbolic representations given by the system* $\mathcal{L}$, *and A the action set. There are k distinct sets of mappings. i denotes the i-th representation generated by the application of operation* $succ(\mathcal{L}_i^k)$ *at the l-th level of successors of the root representation* $C_{0,0}^k$.

For example: Omitting the index k for $k = 1$, the variables and constants $\{a, b, 0, 1000\}$ and operators $\{<, >\}$ make up the set $Z : \{0 < a < 1000, 0 < b < 1000\}$ of formulae in $\mathcal{L}_{0,0}$. The logical connective $\wedge$ defines the set Ω. Thus $C_{0,0} : \mathcal{L}_{0,0} = (0 < a < 1000) \wedge (0 < b < 1000)$ for the initial symbolic representation. The full mapping is described by $R_{0,0} : C_{0,0} \rightarrow \{action_1, action_2\}$

$succ(\mathcal{L}_{0,0})$ is given by

$$\mathcal{L}_{1,1} = (0 < a < 500) \wedge (0 < b < 1000)$$

$$\mathcal{L}_{1,2} = (0 < a < 1000) \wedge (0 < b < 500)$$

$$\mathcal{L}_{1,3} = (500 < a < 1000) \wedge (0 < b < 1000)$$

$$\mathcal{L}_{1,4} = (0 < a < 1000) \wedge (500 < b < 1000)$$

A corresponding successor representation would be denoted is simply $C_{1,1} : \mathcal{L}_{1,1}$, and the mapping written shorthand as $R_{1,1} : C_{1,1} \rightarrow \{action_1, action_2\}$.

This definition will be useful in the following sections and chapters to describe in a compact way the different modes of reasoning that can be implemented with the algorithm.

2.5 Relation to Existing Approaches

To conclude the formal section, BRA is briefly compared put into context with the existing models and methods given in section 2.2.

BRA attempts to provide a mixed approach to learning by combining cognitive aspects (rule extraction) and learning by experience. With respect to the game theory literature, models discussed as mixed models are most closely related. In detail:

- BRA uses the concept of state-space partitioning to balance experimentation and habitualisation. In new situations, agents find out by trial-and-error situational adequate behaviour (if it exists). For known situations, behaviour can become very stable. This is similar to CBR. Gilboa and Schmeidler (1996) find that rules that experiment in unknown cases and tend to habitual repetition in well-known situations are most efficient.

- The update rule in BRA approaches the average reward; a discount parameter determines the speed of this approximation. This can be interpreted as calculating the expected payoff, and is thus similar to the rules used by PA, or in the experiments of Mookherjee and Sopher (Mookherjee and Sopher 1994; 1997).

- Most simple RL and mixed models discussed in this chapter are explicitly designed for (behavioural) game theory. As a computer algorithm,

BRA is more general (rather a framework), and can be applied to any sort of model.

As a computational method, BRA is closely related to CLARION and LCS. As in CLARION and LCS, RL is the most important aspect for generating action-centred knowledge. Differences exist in the way such knowledge is used to build the internal models of the environment:

- BRA does not start with a psychological model of skill acquisition as CLARION or no explicit model at all as machine learning, but a sociopsychological model of bounded rationality.
- BRA uses a pure symbolic representation of conditions with simple first- order predicate logic. CLARION has to transform them in a network structure, LCS in binary strings.
- CLARION modifies rules only after evaluation of bottom level actions; ACS compares prediction errors. BRA is much less sophisticated here, using a simple generate-and-test procedure to decide whether a rule should be specialised or generalised. If the test phase fails (possibly only after a long time when the environment changes), the generated rule is deleted again. CLARION as well as ACS keep detailed statistics and perform complex estimations to decide about generalisation and specialisation of specific rules.
- BRA starts with a state description covering all possible states and builds a model by searching heuristically through the space of these state descriptions that can be expanded logically from the initial descriptor. In CLARION as well as ACS, it is not necessary to describe the state space fully. If new states are encountered, new rules are

created on the fly. BRA is thus much more sensitive to characteristics of the state space. For example, for state variables with large value spaces, specialised rules would be discovered only at later stages of the state expansion mechanism. Even if fine-grained differentiation is useful, they might never be developed because descriptions generated on the path might not be immediately more successful than more general rules, so that the path is not further explored. However, BRA could be extended to cover initially only a small range of conditions, adding new attribute values dynamically as they appear in s.

2.6 An Example

To demonstrate the principle, a simple bargaining game was simulated using the algorithm. The idea of bargaining games is that two players have to agree on a share after a finite number of offerings. If haggling takes too long, both players get nothing. A simplified version of such games with discrete shares is simulated here. In the game, agents can demand a low, medium or high share of a good. Table 2.2 shows the payoffs. This distribution of payoffs leads to situation where demanding a low share guarantees a certain, but low payoff, while demanding a high share may yield a higher, but uncertain payoff.

In the first simulation, there are $N + 1$ agents: $N/2$ agents always demand the highest share, $N/2$ always the lowest. One agent has no predefined strategy, but learns what share to demand from encounters with other players. Agents demanding a low share are green, agents demanding a high share are blue. Each time step, agents are paired randomly and play their strategy. With each encounter the learner is told which colour the opponent has. The agent can then use this information to build the state-action tree.

	low	medium	high
low	0.3 (0.3)	0.3 (0.5)	0.3 (1)
medium	0.5 (0.3)	0.5 (0.5)	0,(1)
high	1,(0.3)	0,(0)	0,(0)

Table 2.2: Payoffs of the demand game. The first number in a cell is the payoff of the row player, the second number the payoff of the column player.

In the second simulation, strategies are assigned randomly to the green and blue agents. Simulations were run with $N = 10$ (i.e. the learner encountered with equal probability a green or blue agent) for 1000 steps. The model parameters were set at $\gamma = 1$, $\zeta = 0.4$, $\rho = 0.3$, $\mu = 25$, $\nu = 19$. The parameter χ is not of interest here, because the question is which categories do emerge; any restriction would be counterproductive. So, χ is set to the arbitrary high value 100. γ is set to 1 to speed up learning since the environment is deterministic; $\zeta = 0.4$, $\rho = 0.3$ are set to moderate values to prevent excessive switching and cycling but still avoiding traps.

The aim of this simulation is to demonstrate the working of the algorithm, not to explain bargaining behaviour. Therefore, only the evolution of the state tree of the learner is analysed.

Figure 2.5 shows the result of the tree-building process: The agent has learnt that it is beneficial to distinguish between the colours of opponents. When it meets green agents, it demands over 80 % of the time a high share of the good, while it demands a low share if blue agents are encountered. The process thus converges to the optimal solution; in most encounters with each type of agent, the maximum payoff is obtained. With only two possible states, this distinction is easy to learn, and consequently discovered early in the simulation. This is shown by the activation frequency of the state descriptors: In just about 50 steps out of 1000 the initial state description 'opponent is blue or opponent is green' is used. The deeper levels 'colour is

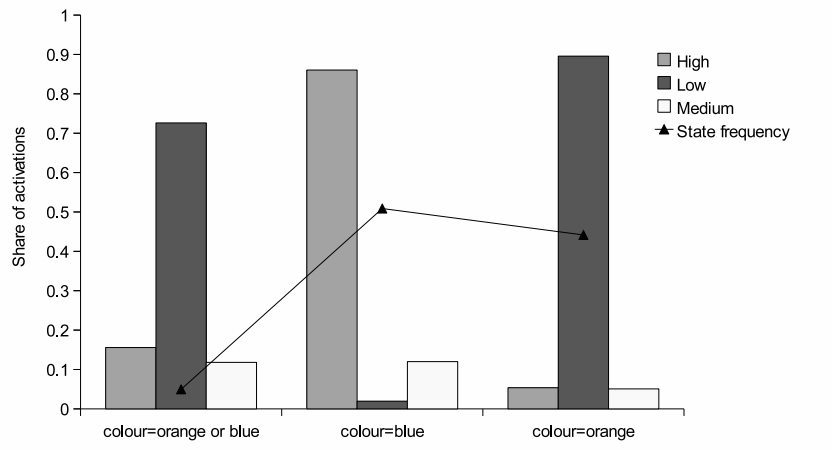


Figure 2.5: BRA example 1. Colours correspond to actual strategies of the agents. The values are fractions of total activations of actions and encounters of state descriptions, respectively.

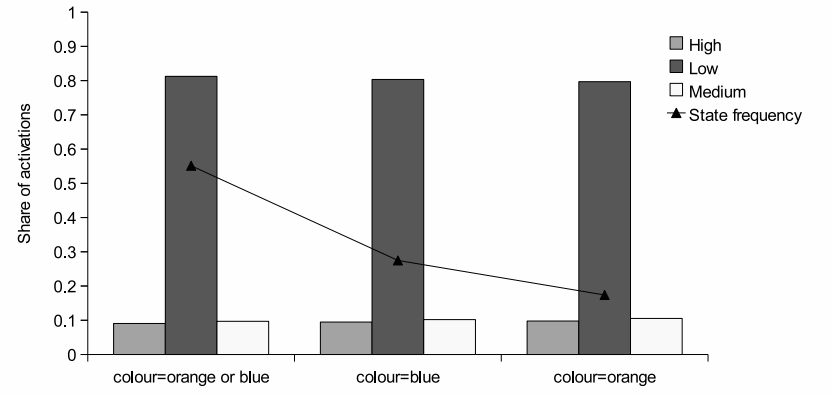


Figure 2.6: BRA example 2. Colours are assigned randomly to strategy types. The values are fractions of total activations of actions, and encounters of state descriptions, respectively.

blue' and 'colour is green' is expanded quickly and remains stable.

Figure 2.6 shows the result if the colours are assigned randomly: Since there is nothing to gain from a distinction of colours, the agent does not pay attention to this attribute. As a result, the agent demands the low share irrespective of the other player's colour 80% of the time. Furthermore, the most frequent state description is the initial state with no differentiation between colours. Thus, the process converges to the dominant strategy. Because it is impossible to use colour as an indicator for the opponent's expected strategy, the learner chooses the action that yields always a positive payoff.

2.7 Conclusion and Outlook

In this chapter, an algorithm aiming to replicate simple decision processes of bounded rational actors has been described, formalised and demonstrated. BRA contributes to reinforcement learning in social simulation and combines elements of approaches already used by CLARION and learning classifier systems. However, it is different from these approaches as it is less general than a cognitive architecture and explicitly built upon a sociopsychological concept of learning. In that sense, the contribution is not the provision of a better or more efficient problem solution method than, e.g., classifier systems. On the contrary, it allows to add cognitive limitations and human mistakes to a learning agent. For an appropriate representation of human learning, BRA can, thus, deliberately be suboptimal (if required). Problem solving methods, however, are typically designed to be efficient. A major difference and advantage to existing learning approaches is furthermore the use of symbolic state representations. This makes a model more tractable than, e.g., a binary string representation or neural network. It

becomes possible to look into the agent's 'mind' and understand its mental model. By the same means, BRA can also cover more abstract concepts in an intuitive way.

The motivation of the example simulations was to assess the performance of the algorithm from a perspective of verification. Being a simple simulation, it was straightforward to verify that the algorithm performed as specified in simple settings. Agents learnt to distinguish simple features in the environment.

BRA is a very general way of representing learning. It attempts to represent learning and bounded rationality in a more realistic way - neither too 'simple' (pure stimulus-response), nor too 'rational' (full information and deliberation). The solution in BRA is to combine a rule-based with an RL-based approach. Being a framework, BRA allows the specification of different learning models. More precisely, the following typical learning cases in ACE scenarios can be represented:

1. Dynamic CBR: The agent learns to behave habitually depending on the situation, without having full knowledge of all possible situations. This is the most general case described by the previous sections, and was demonstrated in the example. More formally, this case can be described with $k \geq 1$, $\bigcap_{k=1}^{n} \mathcal{L}_l^k = \emptyset$, $|A^k| > 1$, and $succ(\mathcal{L}_i^k) \neq \emptyset$. For example $C_{0,0}^1 = (0 < a < 1000)$, $A^1 = \{x, y\}$, $C_{0,0}^2 = (0 < b < 1000)$, $A^2 = \{x, y, z\}$.

2. Static CBR: The agent does not learn rules, but simply learns to behave habitually for a set of given situations. This case can be described with $k \geq 1$, $|A^k| > 1$, $\bigcap_{k=1}^{n} \mathcal{L}_l^k = \emptyset$ for $k > 1$ and $succ(\mathcal{L}_i^k) = \emptyset$. It describes a simple CBR agent who learns optimal actions for a num-

ber of fixed situations. The difference to the previous case is that the successor operation returns an empty set of symbols (i.e. 'nothing').

3. Pure RL: This case is given by further simplification of CBR (2): $k = 1$, $|A^k| > 1$, $\mathcal{L}_0^0 = \emptyset$. There is only one situation, which is described by an empty condition. The agent becomes a simple reinforcement learner like those described in the game theory literature review.

4. Combining CBR and LCS: It is possible to combine the case-based approach of BRA with the classifier idea used in LCS. This can be described with $k \geq 1$, $|A^k| = 1$ and $\bigcap_{k=1}^{n} \mathcal{L}_l^k \neq \emptyset$. *succ* may be empty or non-empty. For example $C_{0,0}^1 = (0 < a < 1000)$, $A^1 = \{x\}$, $C_{0,0}^2 = (0 < a < 1000)$, $A^2 = \{y\}$. Here, several mappings 'compete' to become the current node from which the single action is selected. Initially, the competing mappings are likely to become activated with similar probability (by cycles of generating, testing and deletion of paths). Once true values of the state-descriptors are approached, the agent should eventually apply some rules with higher probability even if the conditions are overlapping. This type of learning is basically a different form of representing case (1) - instead of deciding between action x and y using RL, the state value is used as the decision criterion.

5. Fully deterministic: A BRA agent can become fully deterministic by allowing only one condition and one rule per k. In this case, $k \geq 1$, $|A^k| = 1$, $\bigcap_{k=1}^{n} \mathcal{L}_0^k = \emptyset$ and $succ(\mathcal{L}_0^k) = \emptyset$.

Being a configurable computer simulation framework, the features of the algorithm always depend on the concrete problem modelled. The remaining chapters 3, 4 and 5 are applications of this framework. More specifically,

cases (1), (2) and (3) are represented. Case (4) is, in principle, a different form of case (2) and not further treated. Also, case (5) is not treated since it is not interesting for most models of human behaviour, but belongs to a very subconscious mode of learning where nothing about the environment is known or perceived. In detail, the chapter represent the cases in the following way:

- The statistical discrimination model in chapter 3 is a representative of case (1). The model variants treated have a two-dimensional state space (test results and colours of workers). BRA works on these dimensions, expanding further state descriptions and learns the respective action policies.

- In chapter 4, a simple RL model of network formation is presented, implementing case (2). The cases are given by the player names in the simulations. Learning takes place only in the form of RL; there is no expansion. However, a simple reference model representing case (1) is compared with the simple RL version.

- In chapter 5, learning is further simplified, representing case (3): Patients choose between doctors; no additional cases are needed. From an implementation point of view, RL is realised as case (2) with a single condition - if a consumer is ill he becomes a patient; as a patient he chooses a doctor. To represent this binary choice, a condition is checked in the rule system of the agent before executing the behaviour.

Chapter 3

Statistical Discrimination

3.1 Introduction

Discrimination is the disadvantageous treatment of individuals based solely on their membership of a certain group such as race, age or gender. Economic discrimination occurs in different domains, e.g., in the housing, insurance or labour market. For example, insurance premiums frequently differ among age groups or gender. Women or migrants more often work in jobs below their actual qualification as comparably qualified white males. In labour economics, one speaks of discrimination if members of a certain group who have the same abilities and skills as other groups 'are accorded inferior treatment with respect to hiring, occupational access, promotion, wage rate, or working conditions' (McConnel et al 2006; p.428). Typical forms of discrimination in labour markets are: wage discrimination, where the disadvantaged group receives a lower wage; employment discrimination, where the disadvantaged group is more likely to unemployed; job discrimination where certain groups are restricted from entering certain occupations irrespective of ability; and human capital discrimination, meaning that the disadvantaged group has less access to productivity-increasing opportunities

such as schooling or vocational training (McConnel et al 2006; p.428).

In Economics, Becker (1957) first brought forward a theory of discrimination, which was based solely on preference. He defined employer discrimination as a situation in which employers are prejudiced against a certain group and prefer to employ members of group A but not members of the prejudiced group B (the distinction of A for the advantaged and B for the disadvantaged group will be kept for the remainder of this section). This taste is assigned a monetary value. The strength of this value is called discrimination coefficient d. Employers maximise a utility function that is the sum of profits plus the value of employing members of the particular groups. Prejudiced employers want to hire B workers at a wage rate of w_B. B workers are hired only if their wage is lower to compensate for the discrimination coefficient, thus $w_A = w_B + d$. If the aggregate coefficient d' in the market is sufficiently large, this will create a wage gap between A and B workers as long as labour supply exceeds demand. The model implies that biased employers earn less due to their preferences, as unbiased firms can hire more B workers with equal skills at the lower wage. In the long run, this would eliminate the wage gap, because the number of more profitable, non-discriminatory employers will increase to the point where B workers do not have to work for discriminating employers. In reality, however, differences in wages between groups have mostly persisted.

In the theory of statistical discrimination, on the other hand, inequality between groups arises endogenously. The reason for discriminatory treatment is based on believed or actual average differences between groups. The average characteristic is then ascribed to individual members of each group. When members of the disadvantaged group realise these beliefs and expect to be treated negatively, they may actually adopt this behaviour, which reinforces existing stereotypes.

There are two broad directions of statistical discrimination models. Phelps (1972) and related approaches build models based on exogenously imposed differences. The basis of such models are two groups of workers and employers who observe skills only as a noisy signal, e.g. by using an employment test. Skill and signal are jointly normally distributed. The noisier the signal, a worker's productivity is on average close to the group average. If the signal is precise, it predicts productivity well. Discriminatory outcomes can be generated in basically two ways: Either each group's signal is equally informative, but productivity is different. In this case, one group will receive lower wages as employers expect productivity to be lower. In the other case skills are distributed evenly, but signals are differently informative. Workers belonging to the group with the more informative signal receive higher wages than workers with the same skill belonging to the group with worse signals.

At the same time, Arrow (1973) proposed a model in which initially identical groups can evolve into groups with different productivity due to co-evolving stereotypes on the employer side. In the model, workers invest in human capital conditional on the expected wage. Employers pay a wage depending on the skills of the worker, which they observe perfectly after their hiring. Discrimination can exist if employers expect the skill level of group B to be lower than that of group A. This reinforces wage expectations of the workers of the respective groups. If investing in human capital is not worth the effort for group B, the beliefs of the employers are reinforced, leading to a self-fulfilling prophecy. The result is an equilibrium in which one group does not invest and will consequently be assigned the less well-paid jobs. Coate and Loury (1993) extend this model by making the ex-post skill observation uncertain, which adds higher uncertainty with respect to the observability of workers' actual skill.

Coate and Loury (1993) is the basis for many models, including dynamic approaches and laboratory experiments. Also the model developed in this chapter is based on it and will therefore be described in detail in section 3.2.

The purpose of this chapter is to develop a dynamic model of statistical discrimination in labour markets using BRA as the learning method. There already some dynamic models (e.g. Blume 2006); however they use belief learning methods. Furthermore, the chapter also aims to reproduce the experimental results of Fryer Jr. et al (2005). Using an agent-based model has the advantage that not only the aggregate results, but also individual behaviour can be compared. The research question thus becomes whether and under which conditions statistical discrimination can emerge in an RL model, and whether these mechanisms reflect actual human behaviour.

The outline is as follows: In section 3.2 the theoretical literature is discussed in some detail and in section 3.3 the experimental literature. Emphasis is put on the central approaches: The model of Coate and Loury (1993) (CL); and the laboratory experiment of Fryer Jr. et al (2005), which is based on CL. The RL model is described in section 3.4, which takes the experiment as the starting point for its specification. The model is calibrated for the learning and choice parameters of BRA in section 3.5.1. Then, simulations are run and the dynamics of statistical discrimination analysed in more detail in sections 3.5.2 and .

3.2 Models of Statistical Discrimination

In this section the central approaches of statistical discrimination are discussed. For a more complete, recent review, see Fang and Moro (2011).

In the seminal model of Arrow (1973) groups are ex ante identical; actual differences between groups are derived endogenously. In the model, firms offer two types of jobs, skilled and unskilled. Firms have a production function $f(L_u, L_s)$, where L_s stands for skilled and L_u for unskilled labour. Unskilled workers receive a wage of $w_u = f_1(L_s, L_u)$ and skilled workers a wage of $w_s = f_2(L_s, L_u)$, where f_1 and f_2 denote the first derivative of the first and second arguments of f. Skills are acquired through investment at cost c, which is distributed according to a distribution function $G(\cdot)$, which is independent of worker colour.

The proportion of skilled A workers π_A and skilled B workers π_B is determined by the following process: If a worker is assigned to an unskilled job, he receives w_u, if he is assigned to the skilled group he gets w_s, independent of the colour. The firm conducts a test which determines the skill with certainty. If the worker belongs to the skilled group j, $j \in \{A, B\}$, the employer pays a wage > 0 and 0 otherwise. For this test, the firm must pay a cost r. Arrow claims that competition among firms results in zero profits, so that r can be written as

$$r = \pi_A[f_1(L_s, L_u) - w_A],$$
$$r = \pi_B[f_1(L_s, L_u) - w_B].$$

This implies that

$$w_A = \frac{\pi_B}{\pi_A} w_B + 1 - \left(\frac{\pi_B}{\pi_A} f_1(L_s, L_u) \right).$$

Thus, if $\pi_B < \pi_A$ then $w_B < w_A$, and the resulting segregation between low- and high-skilled jobs can be explained by beliefs instead of preferences.

In equilibrium, the fractions π_A and π_B might differ. Workers invest in skills only if the expected gain exceeds the costs. The gains are given by $w_j - w_u$ for group j workers. The proportion of skilled workers is $G(w_j - w_u)$,

that is the fraction of workers whose investment cost is lower than the wage gain.

Equilibrium is given by

$$\pi_j = G(w_j(\pi_A, \pi_B) - w_u), j \in A, B.$$

While in the symmetric equilibrium $\pi_A = \pi_B$, in the asymmetric $\pi_A \neq \pi_B$. In a situation where most workers of a group invest little, the firms will perceive the group on average as lower-skilled and assign the unskilled job to members of that group. This in turn provides little incentive for the workers to invest in the future, decreasing the average skill of the group. By this mechanism, self-fulfilling prophecies become possible: Because B workers are believed to be not qualified, they invest less so that in the end B workers are indeed less qualified.

The most important difference in Coate and Loury (1993)'s model is that wages are fixed, and that worker skills are not perfectly observable. In the model, firms assign workers of type A and B either to a simple task for which no qualification is required, or a complex task which requires a skill. The wage for the complex task is w, while the wage for the simple task is 0. The firm's return x depends on which task was assigned and the actual qualification: If the worker is qualified and the task complex, the return is $x_q > 0$; if the worker is not qualified and the task complex, the return is $-x_u$. If the task is simple, the return is always 0. Workers decide ex-ante whether to invest in a skill or not. The skill investment cost c is distributed heterogeneous across workers according to a cumulative distribution function $G(\cdot)$. This function is independent of the worker group. $G(c)$ is the fraction of workers with investment costs not greater than c. Firms observe a noisy signal θ of a worker's qualification. The signal is drawn from a uniform interval according to a probability distribution function $f_q(\theta)$ if the

worker is qualified, and $f_u(\theta)$ is the worker is not qualified. f_q and f_u are assumed to satisfy the monotone likelihood ratio property that $l(\theta) \equiv \frac{f_q(\theta)}{f_u(\theta)}$ is strictly increasing and continuous in θ. This implies that workers who invested in skills are more likely to receive a positive signal, and that the ex-post probability that a worker was qualified is also increasing in θ.

The game has three stages. In stage 1, nature draws workers' types and investment cost c. In stage 2, workers make their investment decision given c. As signal $\theta \in [0, 1]$, the firms observe a test result which is drawn from the probability distribution functions f_u or f_q, respectively. In stage 3, firms decide whether to assign workers to the complex or simple task.

A firm will hire a worker only if it believes the worker is qualified. Since the signal is noisy and the probability of being qualified is increasing in θ, a suitable hiring strategy is to set a threshold standard. Workers achieving the standard are assigned to the qualified task, workers who fail to achieve this threshold value are assigned to the unqualified task. More specifically, the posterior probability $\xi(\pi, \theta)$ that a worker is qualified is given by:

$$\xi(\pi, \theta) = \frac{\pi f_q(\theta)}{\pi f_q(\theta) + (1 - \pi) f_u(\theta)} = \frac{1}{1 + \left(\frac{1-\pi)}{\pi}\right) \frac{f_u(\theta)}{f_q(\theta)}}$$

The expected payoff is $\xi(\pi, \theta)x_q - (1 - \xi(\pi, \theta))x_u$. The best policy is to assign the worker to the complex task only if $x_q/x_u \geq (1 - \xi(\pi, \theta)) / (\xi(\pi, \theta))$, which is equivalent to $x_q/x_u \geq (1 - \pi/\pi) (f_u(\theta)/f_q(\theta))$. The threshold $s^*(\pi)$ is given by

$$s^*(\pi) = \min\{\theta \in [0, 1] | \frac{x_q}{x_u} \geq \left(\frac{1 - \pi}{\pi}\right) \frac{f_u(\theta)}{f_q(\theta)}\}.$$

Employers set the standard $s_j = s^*(\pi_j), j \in A, B$, before observing the actual signal. More optimistic beliefs will lead to lower standards, more pessimistic beliefs to higher. Thus, if a group is believed to be less qualified,

investing workers from that group are less likely to get a signal exceeding s^*.

Rational workers invest only if the cost does not exceed the expected benefit. The expected benefit depends on the probability that the worker gets the qualified job, which in turn depends on the standard s^*, and the gross return from the wage of this job. The probability of getting assigned to the qualified job is $1 - F_q(s)$ if the worker invested, and $1 - F_u(s)$ if not. The expected benefit can then be defined as $\beta(s) = \omega\left(F_u(s) - F_q(s)\right)$, where ω is the gross return from being assigned to the complex task. Thus, the worker invests only if $c \leq \beta(s)$. The fraction of workers that become qualified is $G(\beta(s))$. $\beta(s)$ is a single-peaked function of s^*, increasing whenever $f_u(s)/f_q(s) > 1$, and decreasing if $f_u(s)/f_q(s) < 1$, which reflects the monotone-likelihood property. There is little incentive to invest if standards are very high or very low. Either the chance to get the qualified job is always high independent of investment behaviour, or too small to make investment beneficial.

In equilibrium, employers choose standards that induce workers to become qualified at the rate postulated by the beliefs. Formally:

$$\pi_j = G(\beta(s^*(\pi_j))), j \in \{A, B\} \tag{3.1}$$

A discriminatory equilibrium can exist whenever equation 3.1 has multiple solutions. Employers may have the belief that a group is less qualified than the other and consequently, will set higher standards for this group. As this lowers the incentive to invest, the outcome is a self-fulfilling prophecy.

Figure 3.1 shows the equilibrium graphically. Coate and Loury (1993) note that not all solutions of equation 3.1 are locally stable under the implicit adjustment process $\pi^{t+1} = G(\beta(s^*(t^t))$. An initial belief close to π^*

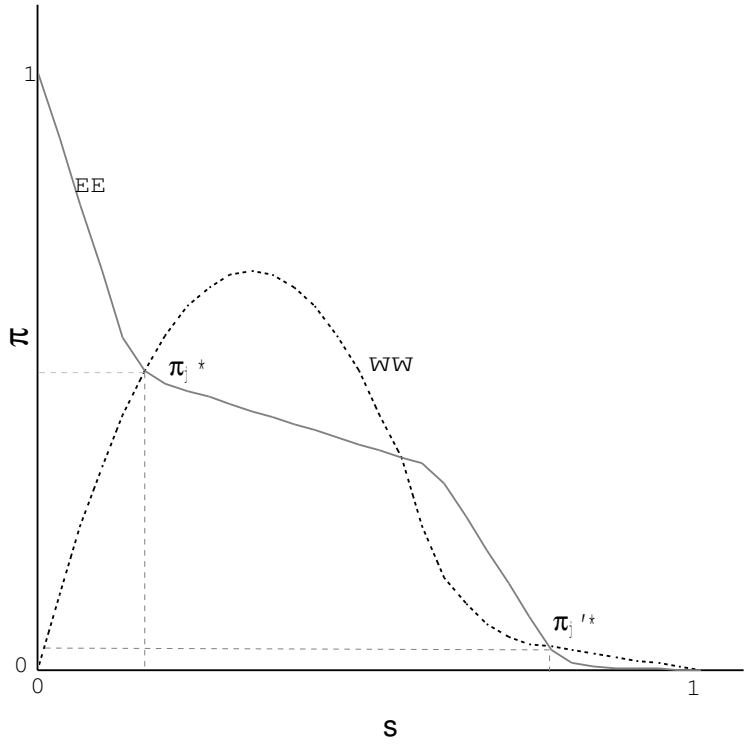


Figure 3.1: Equilibrium in Coate and Loury's statistical discrimination model. The x-axis represents the assignment standards s that need to be fulfilled to be assigned to the qualified task, on the y-axis, π measures the belief how many workers invest in skills. WW depicts pairs of standards and proportions of a group investing consistent with optimal worker behaviour (the graph $\{(s,\pi)|\pi = G(\beta(s))\}$). EE depicts the standard-belief pairs consistent with optimal employer behaviour (the graph $\{(s,\pi)|s = s*(\pi)\}$). A point s and π that lies on both curves solves equation 3.1.

converges only to π^* if the slope of the EE curve exceeds that of WW at π^*. An unstable self-confirming belief is not robust to small errors in employers' perceptions, as the resulting standards will not induce workers to engage into the 'required' investment behaviour.

Coate and Loury (1993) analyse the implications of this model with respect to the question whether discriminatory equilibria can be changed by imposing hiring quotas. They show that there are conditions under

which negative stereotypes can be eliminated. The idea is that in a non-discriminatory equilibrium, an employer assigns the complex task to a randomly selected worker with equal probability. This equality is achieved via an adjustment process of the assignment thresholds $s_j, j \in A, B$ that changes the skill investment incentives of both groups. For qualified workers it becomes more difficult to get assigned to the complex task, while unqualified workers are motivated to increase their skills. Note that in the resulting equilibrium s_A must not necessarily be s_B; the concrete value depends on the initial discriminatory equilibrium. CL illustrate how this process works by the following example: Consider a situation where the employment test can take three outcomes: A pass result, a fail result, and an unclear result. The unclear result corresponds to a signal which can originate both from investing and not-investing workers. Without affirmative action, firms assign with probability 1 workers with bad test results to the unqualified task, and workers with a good result with probability 1 to the qualified task. If the result is unclear, firms can either follow a liberal or conservative strategy. Under the liberal strategy, workers are assigned to the complex task, under the conservative strategy to the simple task. Without intervention, the expected return from the liberal strategy must be large enough to assign the qualified task. If B workers coordinate on the conservative equilibrium because employers have low expectations about B productivity, and A analogously on the liberal equilibrium because of higher expectations, the outcome is discriminatory. If a quota is introduced in such a state, the employer must decide whether to assign more B workers, possibly with a bad test result, to the complex task, or more A workers with ambiguous results to the simple task. If the expected loss of assigning qualified workers to unqualified jobs is greater than the expected gain from assigning unqualified workers to the complex task, the firms will assign all B workers with unclear results, and a fraction of B workers with failed tests to the complex

task, until the employment quota requirement is achieved. The employers thus patronise B workers because the assign them the skilled jobs even though they failed the test. As a consequence, the investment incentives for B workers might be lower as for A workers. Employers continue to view members of group B as less qualified.

Most models of statistical discrimination are static models and state only that discriminatory equilibria might exist. However, how discrimination comes about is not considered and usually attributed to historical circumstances. Only some dynamic models exist, of which the Blume (2006)'s is described in some detail in what follows.

Blume (2006) considers a stochastic model using ideas from evolutionary game theory based on the CL model. There are three types of workers: The common type c can acquire skills at cost $c > 0$; workers of type 0 have no cost of investment; and an 'unteachable' type ∞ with infinite investment costs. The total number of workers is fixed at M, but the size of each subgroup may vary. A worker is of type 0 with probability ρ_0, type ∞ with probability ρ_∞, and of type c with probability $1 - \rho_0 - \rho_\infty$. ρ_0 and ρ_∞ are small. The skill level of the common type is endogenous, while the level of groups 0 and ∞ is fixed at the beginning (always/never skilled). Workers believe to get a skilled job with probability ν. On the employer side, there are two types of firms. Both types value unskilled workers with 0. Type τ firms value a skilled worker at $\tau > 0$, type σ firms with $\sigma > 0$. The probability that a firm is of type σ is ϵ. The cost of hiring an unskilled worker is $\eta > 0$.

Workers have no opportunity to signal their skill; group membership is the only marker. Employers have a common expectation π that a worker is skilled. In each discrete time step, one employer is matched with a worker.

The probability that this happens is given by $q = \min\{N/M, 1\}$. The wage rate for skilled workers is fixed at w; costs are $c < w < \tau$. If a skilled worker is matched with a firm, he earns w; a worker who is not offered a job goes to the unskilled labour market and earns 0.

In equilibrium, workers maximise their expected return with respect to skill acquisition. Firms maximise their profits depending on the expectations about the skill level of the labour force. Type τ firms hire a worker only if expected profits are not negative:

$$\pi\tau - (\pi w + (1 - \pi)\eta) \geq 0$$

The reservation belief π^* that a worker is skilled is given by: $\pi^* = \frac{\eta}{\tau+\eta-w}$; this value makes the firm indifferent about hiring or not. It is assumed for τ-firms that $((1 - \rho_0)\eta + \rho_0 w)/\rho_0 > 0$, which implies $\pi^* > 0$. Similarly, for type-σ firms $((1 - \rho_0)\eta + \rho_0 w)/\rho_0 < \sigma$. From this follows that type σ firms will always hire a worker from the disadvantaged group. Whether type τ firms do so, depends upon its beliefs.

Type c workers believe with probability ν that they will be offered a job, so that the return to skill investment is $\nu w - c$. The reservation belief at which c workers are indifferent whether to acquire skills or not is given by:

$$vw - c = 0$$

The equilibrium is determined by two probabilities: ρ_f, the probability that a type τ firm offers a worker a job, and ρ_w, the probability that a type c workers acquires skills. Thus, equilibrium is a pair ρ_f, ρ_w such that

1. ρ_f maximises $\rho_f(\pi\tau - \pi w - (1 - \pi)\eta)$
2. ρ_w maximises $\rho_w(vw - c)$

3. $\pi = \rho_0 + (1 - \rho_\infty)\rho_w$

4. $\nu = (1 - \epsilon)q\rho_f = \epsilon q$

Hence, analogously to CL, the beliefs π and ν determine the equilibrium. Two possible pure equilibria exist: First, full-employment exists when all workers who can acquire jobs are offered skilled jobs ($\rho_f = 1, \rho_w = 1, \pi = 1, \nu = q$). An underemployment equilibrium is given if type c workers choose not to acquire skills, and only type σ firms offer jobs ($\rho_f = 0, \rho_w = 0, \pi = \rho_0, \nu = q\epsilon$). In the full-employment state, all workers find a job, even the unteachable ones; in the under-employment state, only the workers with zero investment costs get a job. Statistical discrimination exists if both full- and underemployment pure equilibria exist. This happens if $q\epsilon < \nu^*$ assuming that $\tau > w > c, vu^* < q, \rho_0 < \pi^* < 1 - \rho_\infty$. Similarly, if $\nu^* < q\epsilon$ then the only pure equilibrium is the full-employment state. The typical case is $\rho_0 < \pi^* < 1 - \rho_\infty$, as the fractions ρ_0 and ρ_∞ are assumed to be small. This is the basis for the dynamic analysis which is described next.

In the dynamic perspective, I workers enter the labour market at each discrete timestep t. A worker's lifetime has two periods. In the first period at t, they acquire skills. In the second period at $t+1$ they are matched with an employer. Employers hire workers according to their beliefs. Of the M workers at time t, K_t will receive jobs, and J_t of the workers with jobs have in fact skills. Normalising these numbers as fractions defines $k_t = K_t/M$ and $j_t = J_t/M$. From the fractions j_t and k_t firms and all workers update their beliefs to π_{t+1} and ν_{t+1}. All knowledge is public. The newly arrived workers at $t = 1$ make then their skill investment decision based on ν_{t+1}. Since all information is public, workers can predict firms' expectations accurately, so that $v_{t+1} = q$ if $\pi_{t+1} \geq \pi^*$, or $\nu_{t+1} = q\epsilon$ otherwise. If $\pi_t \leq \pi^*$ then only type-σ firms offer jobs, resulting in the underemployment equilibrium

(beliefs in the 'low regime'). If beliefs are $\pi_t \geq \pi^*$, the full-employment equilibrium results (beliefs are in the 'high regime').

The market outcome j_t, k_t is the result of the belief formation in the preceding time steps. The stochastic process $(j_t, k_t)_{t=0}^{\infty}$ thus describes the evolution of the market outcomes. The learning procedure based on public information described above makes the process Markovian with two transition regimes. The probability that π_{t+1} is in low regime depends on the joint distribution of j_t and k_t. Blume shows that there are only two such distributions, leading either the to the high or the low regime. Analysing the long run behaviour reveals that for most parameter values the process remains in one of the two regimes most of the time. More specifically, the parameters π and ϵ determine equilibrium selection. As $\epsilon \to 1$ and $\pi \to 1$ the probability of the high regime goes to 1, and vice versa. As the size of the market grows, the process is more likely to remain in one of the two states permanently.

Blume (2006) discusses a number of policy implications. For example, imposing a hiring quota has the effect of raising ϵ, the probability of being hired in the low regime. If this change is large enough, the underemployment equilibrium will disappear. However, also the opposite might happen, and the probability of the high regime fall to 0: With higher ϵ more workers' true skills are observed, which makes it more difficult to transit from the low-regime if the skill level is low. This is the same conclusion as in Coate and Loury (1993), but there the reason was too low incentives to become qualified; here the reason lies in employers' learning abilities.

Levin (2009) presents a similar stochastic model. The main difference is that time is continuous with workers arriving at a constant flow rate. In the model, employers observe the noisy test signal 'Good' or 'Bad'. As in the

preceding models, workers invest only if expected returns exceed a certain threshold. The process depends on the probability θ of receiving a positive signal. As this probability increases, the process moves to a high regime. Steady states with discriminatory equilibria may evolve. Levin (2009) shows that even if θ increases for the disadvantaged group, this may still not result in higher investment so that negative expectations and discrimination will persist. He also shows that not any increase in θ by, e.g., better access to resources shifts the equilibrium to high state, but only changes that are large enough.

The purpose of this short review was to discuss the major models of statistical discrimination, looking at the dynamics where possible. These concepts provide the basis of the RL model presented in section 3.4. The main features of these models are:

- Discriminatory equilibria can exist if the beliefs of both employers and workers are mutually reinforced.
- Whether a group invests or not depends on employers' beliefs and the probability θ of a positive test result.
- Once discrimination exists, it might take strong interventions to shift the equilibrium from an under-employment state to a full-employment state.

Further extensions of CL or alternative models are not further treated here. For reviews see, e.g., Fang and Moro (2011) or Altonji and Blank (1999). Before presenting the RL model, the next sections looks at statistical discrimination experiments.

3.3 Experiments with Statistical Discrimination

Statistical discrimination has been tested in a series of experiments. Before discussing the experiment of Fryer Jr. et al (2005) in detail, some earlier experiments are summarised based on the review of Anderson et al (2005).

Davis (1987) studied an experimental labour market in which worker groups were of different size. If more observations can be drawn from one group, then it is more likely to produce a higher maximum observation. If employers focus on this higher draw, this may result in a bias towards the larger group. In the experiment, the employer group was in the first period confronted with 80 % of draws from the majority population, in the second they chose themselves how intensively the respective groups should be sampled. Still, 60 % of the employers sampled the majority group, pointing to a mechanism with which a bias towards one group might arise simply induced by population properties (Anderson et al 2005; p.105).

In the experiment of Anderson and Haupert (1999), workers were divided into green and yellow groups. The productivity of each worker in each group was assigned exogenously. Before making a decision, employers could interview the workers at a certain cost. Anderson and Haupert (1999) observed that in markets with lower average productivity of one colour, employers tended to hire fewer workers of that group. They claim that in the absence of an interview, employers focus on the population average. This is supported by the fact that employment levels rose after the cost of interviewing was reduced (Anderson et al 2005; p.106)

Whereas in the previous experiments differences were exogenous, Fryer Jr. et al (2005) conducted a classroom experiment where productivity and hir-

ing decision could evolve simultaneously as in the model of Coate and Loury (1993).

The experiment is set up as follows: Half of the players are employers, the other workers. Half of the workers are green, the other purple. Workers are told that their investment cost is drawn from an interval between \$0 and \$1, and that costs are independent and vary randomly. Workers make their investment decision after observing their cost. After the decision, a test result is generated. If a worker invests and gets hired, he gets a wage of \$3.00. If he is not hired, he gets a low-skill job at a wage of \$1.50. The net gain for an investing worker is the wage minus his investment cost. Two draws of the test are made to determine the final test result. Test results are represented as marbles in an urn. A blue marble (B) represents a positive test result, a red one (R) a negative result. The probability that a result is good is 0.5 if the worker invested, or 0.2 if not. A test result of BB thus means that the chance that a worker invested is high, a result of RR means he probably did not invest, whereas in the event of BR (or RB), the result is unclear. An employer only knows the worker's colour and test result. An employer earns \$4.00 if a worker who invested is hired; \$0.00 if a worker who did not invest was hired, and \$2.00 if the worker was not hired. To both workers and employers, the hiring rates of each colour are presented, i.e. information about the market outcome is public.

Two treatments are presented: In the first treatment, investment costs are drawn for both worker groups over 20 periods from the interval [\$0.00, \$1.00]. The second treatment was conducted to 'investigate the effects of historical discrimination' Fryer Jr. et al (2005; p.166). In this treatment, for the first five periods investment costs for purple workers were drawn from the interval [\$0.5, \$1.00], whereas for green workers from [\$0.00, \$0.5], so that green workers had higher incentives to invest. For the remainder 15

rounds, the cost distributions were equal.

In general, Fryer Jr. et al (2005) observe that discrimination emerges only in some experiments. Of these they present two instances.

In the first experiment discrimination against purple workers emerged quickly. Around 80-90% of green workers were hired most of the time, whereas purple workers were hired at around 40-50%. Hiring rates remained almost constant for green, and slightly improved for purple workers. Investment rates for both groups increased for some periods, after which they fell again. Investment costs in the first two rounds was (by chance) higher for purple workers. This 'may have been a factor that kept investment rates much higher for green workers in most periods' (Fryer Jr. et al (2005; p.165)). Employers hired always when the test result was BB. Employers were more liberal with green workers: If the test result was unclear, they were hired invariably, but only 78% of purple workers. If the result was RR, employers still hired 64% of green, but only 15% of purple workers. In the following discussion, it emerged that beliefs that purple workers would not invest formed quickly, as well as the corresponding belief that this group is unlikely to get hired. This lead most workers of that colour to decrease their efforts. Moreover, the consistent liberal treatment of green workers encouraged most of them in their investment behaviour, while some players stopped investing because they expected to get hired anyway. Thus, investment rates for both groups declined in the second half of the game, but for different reasons.

In the second experiment, it emerged that investment rates of green and purple workers were similar, although the costs for purple workers were much higher. They were hired at an only slightly lower rate than green workers. After step 5, the cost distributions became equal again. Pur-

ple workers continued to invest at similar rates, while investment rates for greens dropped quickly, resulting in higher employment for purples (raising from about 60% to 90%), and lower employment for greens (decreasing from about 65% to 50%).

Summarising, these results highlight some driving factors in experimental environments:

- Negative stereotypes can form quickly and are persistent. It might only take some random perturbations (here, initial cost asymmetries) to generate these stereotypes.
- Decisions are not independent. The belief that one group is more productive leads to the belief that this still holds if bad or mixed outcomes occur, while the opposite is true for the disadvantaged group.
- The height of the cost does not necessarily have a large impact on the investment decisions, as long as the return to investment is positive.

3.4 A Reinforcement Learning Model of Statistical Discrimination

There are n worker agents and m employer agents. Workers are assigned the colours green and purple with equal probability. Each round, workers and employers are paired randomly. Employers must decide to hire or not to hire a worker depending on the result of an employment test and the colour of the worker. If no investment is made, the worker incurs no cost. The test outcome might be either good (+) or bad (-). Two draws are made. The probability of a positive test signal are drawn from two distributions; $f_q(\theta)$ if the worker invested, and $f_u(\theta)$ if not. Table 3.1 shows the payoffs. In the

following simulations, investment cost c is fixed at 0.1 throughout, so that there is never a negative payoff.

	hire	not hire
invest	0.3-c (0.4)	0.15-c (0.2)
not invest	0.3 (0)	0.15 (0.2)

Table 3.1: Payoffs for the RL statistical discrimination model (employer payoffs in brackets)

This model setup is with minor variations identical to Fryer Jr. et al (2005). The main difference is that all information is private. Employers and workers have no information about the employment levels of the respective groups as in the original game. Another difference is the magnitude of rewards, which was divided by 10 for this experiment (simply to standardise values between 0 and 1), and the distribution of players. Furthermore, in the Fryer experiment, there were as many employers as workers and workers were split exactly half in green, and half in purple. In the simulation, workers' groups are partitioned randomly, so that one worker group is often larger than the other. Moreover, there are only half as many employers as workers. The reason is to support learning: The smaller the worker group, the more likely there will be similar behaviour simply by chance, and thus it is 'easier' to discriminate. Similarly, with fewer employers, variation may decrease by chance, and feed back into worker decisions As a consequence, only half of the worker population is matched each round, while all employers act. However, as the simulation results below will illustrate, this seems not to be necessary for generating discrimination.

Using the BRA approach developed in chapter 2, the agents are implemented as follows: Workers have a simple state-action mapping with an empty state description and invest/not-invest as action set. Employers, on

the other hand, may use the different test outcomes and colours of the agents to construct rules according to BRA. The action set consists of hire/not-hire. Different ways are possible to generate rules that constitute beliefs about the relationship between colour and productivity. If the majority of generated rules are based on colour alone, statistical discrimination is clearly observable; if rules are only based on test only, there is meritocracy.

Three different setups are considered. Using the convention of definition 4, they can be described as follows:

Variant I In this variant, there is only limited learning. Only if test results are ambiguous (+- or -+; since both events are equivalent +- is used as the representative for both from here on) agents may learn; otherwise employers always hire if the result is good (++), or never hire if the result is bad (- -). This corresponds to the example constructed by Coate and Loury (1993) described above. Employers can learn a conservative or liberal strategy, depending on their beliefs about the productivity of each group - if in doubt they can either believe that the test comes from a productive worker or the opposite. The deterministic rules can be described by $r^1_{0,1}$: $C^1_{0,1}$ →hire with $C^1_{0,1}$:(test-result = ++) and $r^2_{0,1}$: $C^2_{0,1}$ →not-hire with $C^2_{0,1}$:(test-result = - -). The corresponding initial state-action mapping for the learning problem is $r^3_{0,1} : C^3_{0,1} \rightarrow A$ with $C^3_{0,1}$: (test-result = +-) *and* (colour = purple *or* colour = green). Using definition 4, the decision model can hence be described by $k = 3$, $|A^k| = 2$, $\bigcap_{k=1}^{3} \mathcal{L}_l^k = \emptyset$ and the symbols in table 3.2. The table describes all possible rules the search process can expand.

Variant II In this variant, three different sets of state-action mappings are specified. The first set contains only one initial rule $r^1_{0,1} : C^1_{0,1} \rightarrow A$ with $C^1_{0,1}$: (test-result = ++) *and* (colour = purple *or* colour = green), the

$\mathcal{L}_0^1 = \{\text{test-result} = \text{- -}\}$
$\mathcal{L}_0^2 = \{\text{test-result} = \text{++}\}$
$\mathcal{L}_0^3 = \{(\text{test-result} = \text{+-} \wedge (\text{colour} = \text{purple} \vee \text{colour} = \text{green}))\}$
$\mathcal{L}_1^1 = \mathcal{L}_1^2 = succ(\mathcal{L}_0^1) = succ(\mathcal{L}_0^2) = \emptyset$
$\mathcal{L}_{1,1}^3 = succ(\mathcal{L}_0^3) = \{\text{test-result} = \text{+-} \wedge \text{colour} = \text{purple}\}$
$\mathcal{L}_{1,2}^3 = succ(\mathcal{L}_0^3) = \{\text{test-result} = \text{+-} \wedge \text{colour} = \text{green}\}$
$succ(\mathcal{L}_2^3) = \emptyset$

Table 3.2: Description of all possible rules in Model Variant I. The events +- and -+ are summarised as +-.

second set contains $r_{0,1}^2 : C_{0,1}^2 \rightarrow A$ with $C_2^{0,1}$: (test-result = - -) *and* (colour = purple *or* colour = green), and the third set is given by $r_{0,1}^3 : C_{0,1}^3 \rightarrow A$ with $C_{0,1}^3$: (test-result = +-) *and* (colour = purple *or* colour = green).

Based on definition 4, the model can be decribed by $k = 3$, $|A^k| = 2$, $\bigcap_{k=1}^{3} \mathcal{L}_l^k = \emptyset$ and the symbols described in table 3.3, which again describes all possible mappings.

This specification pre-wires some knowledge about the relationship between test-result and productivity by restricting the possible combinations of the condition elements. Per mapping, only two rules can be generated, limiting the maximum number of rules to six. Subjects observe test result and colour, and based on the test result they start deliberating how to treat the worker from the respective groups.

Variant III This variant finally poses the most challenging learning task. The initial rule can be described correspondingly with $r_{0,1}^1 : C_{0,1}^1 \rightarrow A$ with $C_{0,1}^1$: (test-result = ++ or test-result = - - or test-result = +-) *and* (colour = purple *or* colour = green). Agents start with no prior knowledge or categories at all.

$\mathcal{L}_0^1 = \{(\text{test-result} = ++ \wedge (\text{colour} = \text{purple} \vee \text{colour} = \text{green}))\}$

$\mathcal{L}_0^2 = \{(\text{test-result} = \text{-} \wedge (\text{colour} = \text{purple} \vee \text{colour} = \text{green}))\}$

$\mathcal{L}_0^3 = \{(\text{test-result} = +\text{-} \wedge (\text{colour} = \text{purple} \vee \text{colour} = \text{green}))\}$

$\mathcal{L}_{1,1}^1 = succ(\mathcal{L}_0^1) = \{\text{test-result} = ++ \wedge \text{colour} = \text{purple}\}$

$\mathcal{L}_{1,2}^1 = succ(\mathcal{L}_0^1) = \{\text{test-result} = ++ \wedge \text{colour} = \text{green}\}$

$\mathcal{L}_{1,1}^2 = succ(\mathcal{L}_0^2) = \{\text{test-result} = \text{- -} \wedge \text{colour} = \text{purple}\}$

$\mathcal{L}_{1,2}^2 = succ(\mathcal{L}_0^2) = \{\text{test-result} = \text{- -} \wedge \text{colour} = \text{green}\}$

$\mathcal{L}_{1,1}^3 = succ(\mathcal{L}_0^3) = \{\text{test-result} = +\text{-} \wedge \text{colour} = \text{purple}\}$

$\mathcal{L}_{1,2}^3 = succ(\mathcal{L}_0^3) = \{\text{test-result} = +\text{-} \wedge \text{colour} = \text{green}\}$

$succ(\mathcal{L}_1^1) = \emptyset;$

$succ(\mathcal{L}_1^2) = \emptyset$

$succ(\mathcal{L}_1^3) = \emptyset$

Table 3.3: Description of all possible rules in Model Variant II. The events +- and -+ are summarised as +-.

Based on definition 4, this variation of the model can be described with $k = 1$, $|A^k| = 2$ and the symbols in table 3.4.

$\mathcal{L}_0^1 = \{(\text{test-result} = ++ \vee \text{test-result} = \text{- -} \vee \text{test-result} = +\text{-})$
$\wedge (\text{colour} = \text{purple} \vee \text{colour} = \text{green})\}$

$\mathcal{L}_{1,1}^1 = succ(\mathcal{L}_0^1) = \{(\text{test-result} = ++) \wedge (\text{colour=green} \vee \text{colour} = \text{purple})\}$

$\mathcal{L}_{1,2}^1 = succ(\mathcal{L}_0^1) = \{(\text{test-result} = +\text{-}) \wedge (\text{colour=green} \vee \text{colour} = \text{purple})\}$

$\mathcal{L}_{1,3}^1 = succ(\mathcal{L}_0^1) = \{(\text{test-result} = \text{- -}) \wedge (\text{colour=green} \vee \text{colour} = \text{purple})\}$

$\mathcal{L}_{1,4}^1 = succ(\mathcal{L}_0^1) = \{(\text{test-result} = \text{- -} \vee \text{test-result} = +\text{-})$
$\wedge (\text{colour=green} \vee \text{colour} = \text{purple})\}$

$\mathcal{L}_{1,5}^1 = succ(\mathcal{L}_0^1) = \{(\text{test-result} = ++ \vee \text{test-result} = \text{- -})$
$\wedge (\text{colour=green} \vee \text{colour} = \text{purple})\}$

$\mathcal{L}_{1,6}^1 = succ(\mathcal{L}_0^1) = \{(\text{test-result} = +\text{-} \vee \text{test-result} = ++)$
$\wedge (\text{colour=green} \vee \text{colour} = \text{purple})\}$

$\mathcal{L}^1_{1,7} = succ(\mathcal{L}^1_0) = \{(\text{test-result} = ++ \vee \text{test-result} = \text{- -} \vee \text{test-result} = \text{+-})$
$\wedge\ (\text{colour} = \text{purple})\}$
$\mathcal{L}^1_{1,8} = succ(\mathcal{L}^1_0) = \{(\text{test-result} = ++ \vee \text{test-result} = \text{- -} \vee \text{test-result} = \text{+-})$
$\wedge\ (\text{colour} = \text{green})\}$
$\mathcal{L}^1_{2,1} = succ(\mathcal{L}^1_{1,1}) = \{\text{test-result} = ++ \wedge \text{colour} = \text{green}\}$
$\mathcal{L}^1_{2,2} = succ(\mathcal{L}^1_{1,1}) = \{\text{test-result} = ++ \wedge \text{colour} = \text{purple}\}$
$\mathcal{L}^1_{2,3} = succ(\mathcal{L}^1_{1,2}) = \{\text{test-result} = \text{+-} \wedge \text{colour} = \text{green}\}$
$\mathcal{L}^1_{2,4} = succ(\mathcal{L}^1_{1,2}) = \{\text{test-result} = \text{+-} \wedge \text{colour} = \text{purple}\}$
$\mathcal{L}^1_{2,5} = succ(\mathcal{L}^1_{1,3}) = \{\text{test-result} = \text{- -} \wedge \text{colour} = \text{green}\}$
$\mathcal{L}^1_{2,6} = succ(\mathcal{L}^1_{1,3}) = \{\text{test-result} = \text{- -} \wedge \text{colour} = \text{purple}\}$
$\mathcal{L}^1_{2,7} = succ(\mathcal{L}^1_{1,4}) = succ(\mathcal{L}^1_7) =$
$\{(\text{test-result} = \text{- -} \vee \text{test-result} = \text{+-}) \wedge \text{colour} = \text{green}\ \}$
$\mathcal{L}^1_{2,8} = succ(\mathcal{L}^1_{1,4}) = succ(\mathcal{L}^1_8) =$
$\{(\text{test-result} = \text{- -} \vee \text{test-result} = \text{+-}) \wedge \text{colour} = \text{purple}\}$
$\mathcal{L}^1_{2,9} = succ(\mathcal{L}^1_{1,5}) = succ(\mathcal{L}^1_8) =$
$\{(\text{test-result} = ++ \vee \text{test-result} = \text{- -}) \wedge (\text{colour} = \text{purple})\}$
$\mathcal{L}^1_{2,10} = succ(\mathcal{L}^1_{1,5}) = succ(\mathcal{L}^1_7) =$
$\{(\text{test-result} = ++ \vee \text{test-result} = \text{- -}) \wedge (\text{colour} = \text{green})\}$
$\mathcal{L}^1_{2,11} = succ(\mathcal{L}^1_{1,6}) = succ(\mathcal{L}^1_8) =$
$\{(\text{test-result} = \text{+-} \vee \text{test-result} = ++) \wedge (\text{colour} = \text{purple})\}$
$\mathcal{L}^1_{2,12} = succ(\mathcal{L}^1_{1,6}) = succ(\mathcal{L}^1_7) =$
$\{(\text{test-result} = \text{+-} \vee \text{test-result} = ++) \wedge (\text{colour} = \text{green})\}$
$\mathcal{L}^1_{2,13} = succ(\mathcal{L}^1_{1,7}) = \{\text{test-result} = \text{+-} \wedge \text{colour} = \text{green}\}$
$\mathcal{L}^1_{2,14} = succ(\mathcal{L}^1_{1,8}) = \{\text{test-result} = \text{+-} \wedge \text{colour} = \text{purple}\}$
$succ(\mathcal{L}^1_{2,1}) = succ(\mathcal{L}^1_{2,2}) = succ(\mathcal{L}^1_{2,3}) = succ(\mathcal{L}^1_{2,4}) = succ(\mathcal{L}^1_{2,5}) =$
$succ(\mathcal{L}^1_{2,6}) = succ(\mathcal{L}^1_{2,3}) = succ(\mathcal{L}^1_{2,13}) = succ(\mathcal{L}^1_{2,3}) = succ(\mathcal{L}^1_{2,14}) = \emptyset$
$\mathcal{L}^1_{3,1} = succ(\mathcal{L}^1_{2,7}) = \mathcal{L}^1_{2,3}$
$\mathcal{L}^1_{3,2} = succ(\mathcal{L}^1_{2,7}) = \mathcal{L}^1_{2,5}$

$\mathcal{L}^1_{3,3} = succ(\mathcal{L}^1_{2,8}) = \mathcal{L}^1_{2,4}$
$\mathcal{L}^1_{3,4} = succ(\mathcal{L}^1_{2,8}) = \mathcal{L}^1_{2,6}$
$\mathcal{L}^1_{3,5} = succ(\mathcal{L}^1_{2,9}) = \mathcal{L}^1_{2,2}$
$\mathcal{L}^1_{3,6} = succ(\mathcal{L}^1_{2,9}) = \mathcal{L}^1_{2,6}$
$\mathcal{L}^1_{3,7} = succ(\mathcal{L}^1_{2,10}) = \mathcal{L}^1_{2,1}$
$\mathcal{L}^1_{3,8} = succ(\mathcal{L}^1_{2,10}) = \mathcal{L}^1_{2,5}$
$\mathcal{L}^1_{3,9} = succ(\mathcal{L}^1_{2,11}) = \mathcal{L}^1_{2,2}$
$\mathcal{L}^1_{3,10} = succ(\mathcal{L}^1_{2,11}) = \mathcal{L}^1_{2,4}$
$\mathcal{L}^1_{3,11} = succ(\mathcal{L}^1_{2,12}) = \mathcal{L}^1_{2,1}$
$\mathcal{L}^1_{3,12} = succ(\mathcal{L}^1_{2,12}) = \mathcal{L}^1_{2,3}$
$succ(\mathcal{L}^1_3) = \emptyset$

Table 3.4: Description of all possible rules in Model Variant III. The events +- and -+ are summarised as +-.

3.5 Simulations

Simulations are run in three steps:

1. First, the model is explored to find the RL parameter settings for α and γ that are capable of generating discrimination. The other parameters are fixed. The results of the optimisation procedure are looked at in some illustrative simulations.

2. Using the results of these exploratory simulations, the RL parameters are fixed and more simulations with a larger number of agents and more repetitions are run. The key statistical discrimination parameter θ is varied. The aim is to obtain representative samples of the model behaviour.

3. Some more specific scenarios modelling existing negative stereotypes, taste-based discrimination and variation in investment costs are run to analyse further which features are responsible for generating statistical discrimination.

3.5.1 Exploration

The aim of this section is to find out which model setups under which parameter settings are able to generate statistical discrimination. For this, many simulations with few agents and many α and γ parameter settings are run and optimised using a Genetic Algorithm (GA) as a stochastic optimisation method. The following paragraphs describe the procedure used and the outcome of these simulations.

3.5.1.1 Finding Optimal Learning Parameters

GA's (Holland 1975) are often used in stochastic optimisation problems. An optimisation problem is, for instance, the approximation of a function that estimates some empirical observable value. GA's are, in principle, a directed search process. At the start of the process, a pool of candidate solutions called chromosomes is initialised. A chromosome contains a number of genes. The genes represent, for example, parameter values of a function to be approximated. A chromosome is initialised with a number of typically randomly initialised genes. A gene is represented as a binary string. The bits of the string can encode different things such as the digits of a number. The task of the algorithm is to evolve and select the best solutions from the chromosome pool by applying genetic operators such as mutation or crossover. These operators change and recombine the bits of the fittest genes. Mutation switches a bit of the string with a certain probability; crossover selects a fraction of the chromosomes and recombines genes of

the same type from the resulting sample at a randomly selected point in the string. The new pool of chromosomes constitutes the next generation. Fitness is determined by a fitness function, which computes the distance from the candidate solution to the problem solution. While a subset of the fittest genes is reproduced in the next generation using the operators, unfit genes are removed from the population. The process stops after a certain criterion, e.g., after a maximum number of generations has been computed, or some fitness threshold has been reached (for an introduction, see, for example, Goldberg (1989).

Here, the GA is initialised with chromosomes containing four genes. The genes represent the parameters $\alpha_{employer}$, $\gamma_{employer}$, α_{worker} and γ_{worker}. The population size of a generation is limited to 20 chromosomes; the maximum number of evolutions is bound to 25. The probability that mutation occurs is set at 0.35. The crossover rate is 12%. The framework used for implementation is JGAP (2011). The fitness function is given by the resulting employment discrimination after a simulation run of 5000 time steps. For this, the average difference between employment levels between green and purple workers over all time steps is computed. The larger the difference, the 'fitter' the candidate parameter set. The time scale has no relation to the classroom game. The reason is that this model is of an exploratory nature and unknown whether the necessary agent learning can be achieved in 'real time'. Moreover, large time scales can inform about the stability of the model in the long run.

Applying this algorithm to each model variant produces for each variant a set of different optimal parameter values. Before looking at some example runs in the next sections, the parameters and the outcome of the optimisation procedure are shown.

Table 3.5 summarises the relevant parameters for each model variant. χ is set to an arbitrary high value (100 in this case), because the goal is to find out whether it is possible to generate discrimination at all. Therefore, a limitation of state descriptions provides no benefit at this stage. ζ is set to a small value to allow frequent re-evaluation of mappings and consequently, adjustment in the beliefs. That is, it is easier to revise negative stereotypes. If discriminatory outcomes emerge, they are likely to be based on co-evolution and not ignorance on the employer side. Similarly, the parameters ν and μ are set at intervals that allow reasonable large samples of rewards for single rules (about 100), but can be changed frequently enough over the 5000 time steps to allow reasonable variation in the expanded state-action mappings. Finally, ρ was fixed at a value > 0 to prevent traps in the search process, but not too large to prevent excessive switching.

Parameter	value	meaning
Discrimination parameters		
$f_q\theta$	0.5	Probability of good test result if invested
$f_u\theta$	0.2	Probability of good test result if not invested
c	0 - 0.1	Investment cost interval
BRA parameters		
ζ	0.05	Weight for revisiting expanded nodes
ρ	0.3	Weight for switching paths in the tree
μ	75	Interval for deleting inferior expansions
ν	100	Interval for creating new expansions
χ	100	Maximum numbers of nodes
Variant I		
$\alpha_{employer,r^1}$	0.01 - 0.15	choice parameter for r^1 - action set bound to descriptor $\mathcal{L}_0^1$ (test-result is ambiguous and (colour is green or colour is purple))
$\gamma_{employer,r^1}$	0.01 - 0.5	discount parameter for r^1
α_{worker}	0.01 - 0.2	choice parameter for worker rule
γ_{worker}	0.01 - 0.5	discount parameter for worker rule
Variant II		
$\alpha_{employer,r^1}$	0.01 - 0.1	choice parameter for r^1 - action set bound to descriptor $\mathcal{L}_0^1$ ((test-result=+-) and (colour=green or colour=purple))
$\gamma_{employer,r^1}$	0.01 - 0.5	discount parameter for r^1
$\alpha_{employer,r^2}$	0.01 - 0.1	choice parameter for r^2 - action set bound to descriptor $\mathcal{L}_0^2$ ((test-result= ++) and (colour=green or colour=purple))
$\gamma_{employer,r^2}$	0.01 - 0.5	discount parameter for r^2
$\alpha_{employer,r^3}$	0.01 - 0.1	choice parameter for r^3 - action set bound to descriptor $\mathcal{L}_0^3$ ((test-result=- -) and (colour=green or colour=purple))
$\gamma_{employer,r^3}$	0.01 - 0.5	discount parameter for r^3
α_{worker}	0.01 - 0.2	choice parameter for worker rule
γ_{worker}	0.01 - 0.5	discount parameter for worker rule
Variant III		
$\alpha_{employer,r^1}$	0.01 - 0.15	choice parameter for r^1 - action set bound to descriptor $\mathcal{L}_0^1$ ((test-result=++ or test-result=+- or test-result=- -) and (colour=reen or colour=purple))
$\gamma_{employer,r^1}$	0.01 - 0.5	discount parameter for r^1
α_{worker}	0.01 - 0.2	choice parameter for worker rule
γ_{worker}	0.01 - 0.5	discount parameter for worker rule

Table 3.5: Simulation parameters for finding optimal RL parameters.

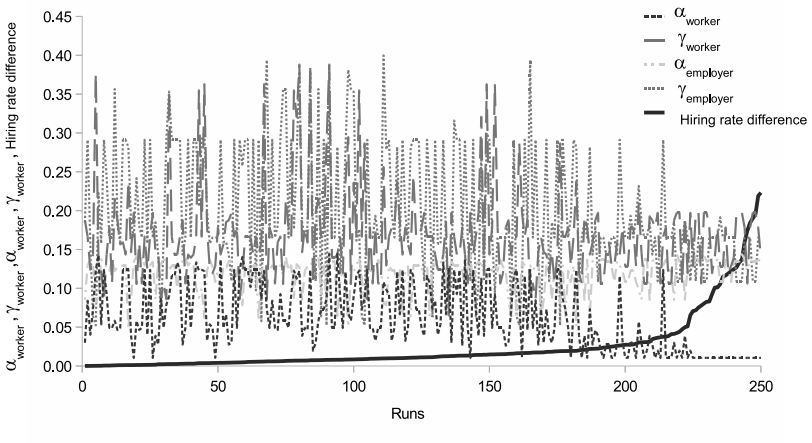


Figure 3.2: Model fit for various parameter settings in model variant I.

Figures 3.2 to 3.4 summarise the different parameter settings that have been visited by the GA. The figures display all samples which were run and are sorted by the largest difference in the employment levels of the two groups. The x-axis represent simulation runs while the y-axis displays the various parameters of the model and the fitness criterion, which all vary between 0 and 1. On the right end of the graph are those simulations that produce the strongest discrimination.

In general, settings with small α_{worker} (early lock-in into investment/non-investment behaviour) have the best chances to produce the discrimination. Thus, if worker behaviour is relatively stable and differs, for some reason, between groups (here due to the variation of choice and learning behaviour), then employers discriminate between them accordingly.

3.5.1.2 Variant I

In this scenario, learning happens only when test results are ambiguous. In the case of - - employers never hire, in the case of ++ they always hire.

Figure 3.2 shows that discriminatory outcomes emerge mostly in case

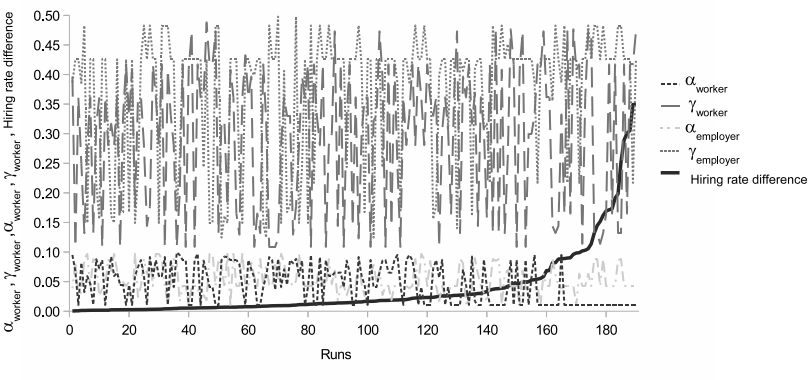


Figure 3.3: Model fit for various parameter settings in model variant II

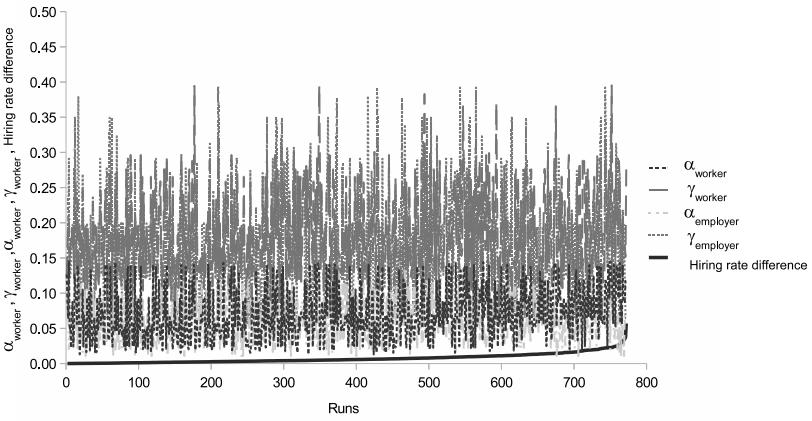


Figure 3.4: Model fit for various parameter settings in model variant III.

where workers' α is small. The reason for this is that the workers act as an 'environment' for the employers. The more stable workers' behaviour, the easier it becomes for the employers to learn. When the initial configuration is such that sufficient members of each group play almost exclusively one of the two possible strategies, discrimination can emerge very quickly and remain stable. As soon as workers' α increases, the employers are increasingly unable to use colour as a decision hint, and the hiring rates for both groups become similar.

The sample simulation shown in figures 3.5 and 3.6 illustrates this dynamic. Discrimination persists, but investment behaviour of the workers is

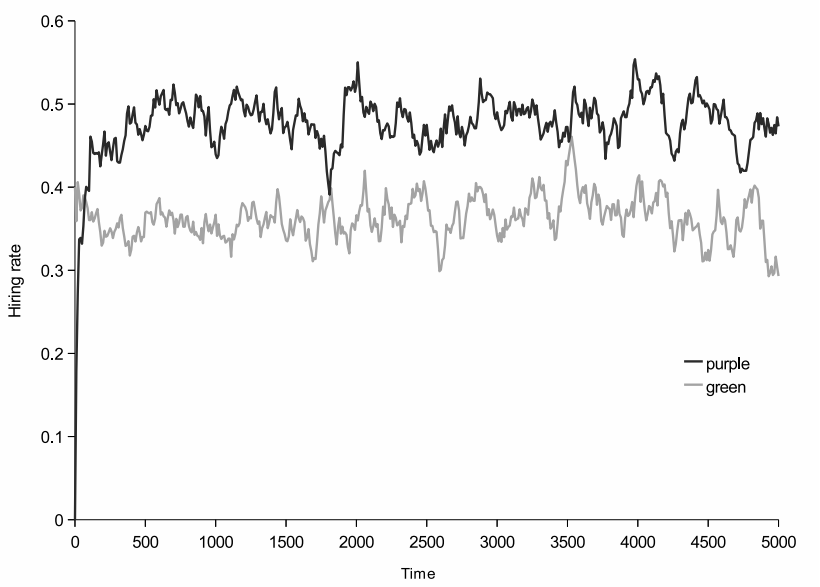


Figure 3.5: Hiring rates of an example run for model variant I (5 purple workers, 5 green workers). Parameters: $\alpha_{employer} = 0.13$, $\gamma_{employer} = 0.17$, $\alpha_{worker} = 0.01$, $\gamma_{worker} = 0.17$

non-deterministic. There were also simulations with even greater discrimination; however in these simulation workers never changed their strategies, i.e. no experimentation occurred any more, and the result was determined fully by the matching of the first time step.

The sample simulation generates discrimination, but stays short of the simulation reported by Fryer et al.: In both cases, there is persistent discrimination, and this behaviour emerges very early. However, in the classroom experiment, discrimination was, with employment rates of 0.8 and 0.4, much larger. In the simulation, rates are on average about 0.48 and 0.36, in the most extreme case the rates reached 0.55 and 0.24. A simple calculation reveals that the hiring rates in the simulations reflect a liberal strategy independent of worker colour: For investing workers, the test outcome probabilities are $++ = 0.5 * 0.5 = 0.25$, $-- = 0.5 * 0.5 = 0.25$, and $+- = 2(0.5 * 0.5) = 0.5$ (since event $+-$ and $-+$ are equivalent,

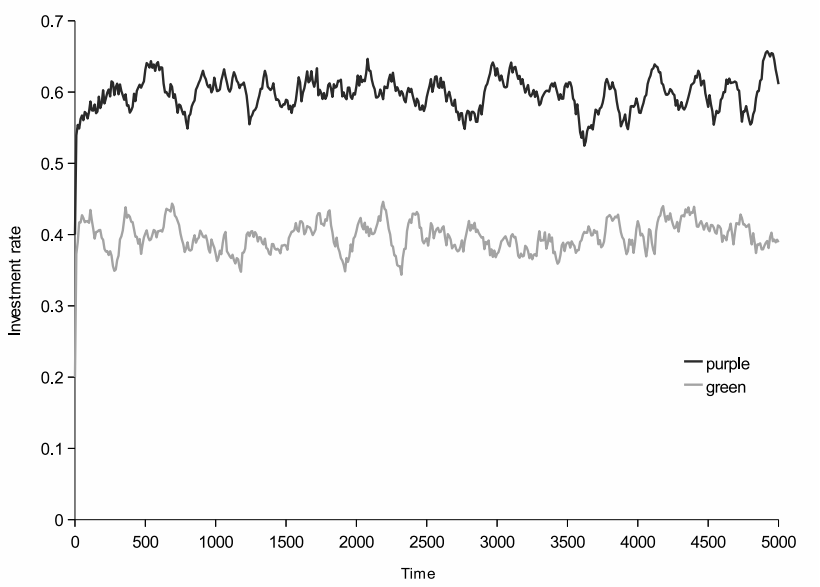


Figure 3.6: Investment rates of an example run for model variant I (5 purple workers, 5 green workers). Parameters: $\alpha_{employer} = 0.13$, $\gamma_{employer} = 0.17$, $\alpha_{worker} = 0.01$, $\gamma_{worker} = 0.17$

they are denoted only with $+-$); for non-investing workers these probabilities are $++ = 0.02 * 0.02 = 0.04$, $-- = 0.8 * 0.8 = 0.64$, and $+- = 2(0.016 * 0.016) = 0.032$. In the simulation, purple workers invest with a relative frequency of 0.6 and green workers with a relative frequency 0.4. In case of a liberal strategy independent of colour (represented by the rule 'if test result is ambiguous, always hire'), the expected employment level is $0.6 * 0.75 = 0.45$ for purple workers, and $0.4 * 0.75 = 0.3$ for green workers. This is, by and large, reflected by the simulation results. Employers hire fewer green workers because their test results are usually worse. If employers were biased against green workers because they invest less, the employment rate of greens should be lower (for a pure conservative strategy, the rate would be 0.1 only).

This means that difference in employment levels can be generated and persist in this model. However, it seems likely that it is not the type of

discrimination observed in the classroom game, where decisions were biased towards 'hire' in case the worker belonged to the group with higher expected investments and towards 'not-hire' in the opposite case. In the RL model, discrimination reflects actual differences in worker productivity.

3.5.1.3 Variant II

The example run of model variant I indicates that employer decisions reflect actual investment behaviour. This resulted into a colour-independent liberal strategy. Possibilities for discrimination were limited, as employers could use colour as decision criterion only if the test result was ambiguous. In the second setup, employers have the capability to favour workers even if their result was bad, and to discriminate even if the test result was good.

Looking at the fitness of simulations with different parameter settings, figure 3.3 shows analogous behaviour as Variant I.

However, the employment level is higher and as figure 3.7 shows, the difference in employment levels is much stronger. Moreover, it seems that this equilibrium state can collapse quickly for no or only very little changes in investment behaviour (figure 3.8).

This result comes closer to the empirical results of Fryer Jr. et al (2005), who observed that if in doubt, workers expected to invest are hired at a higher rate than if expected not to invest. Discrimination is persistent; however, the level of employment and the extent of discrimination may change quickly. As it can be seen from figure 3.8, this is due to a preceding change in investment behaviour.

Calculating the hiring rates for a liberal strategy in the same way as in variant I results in expected employment levels of 0.44 for purple and

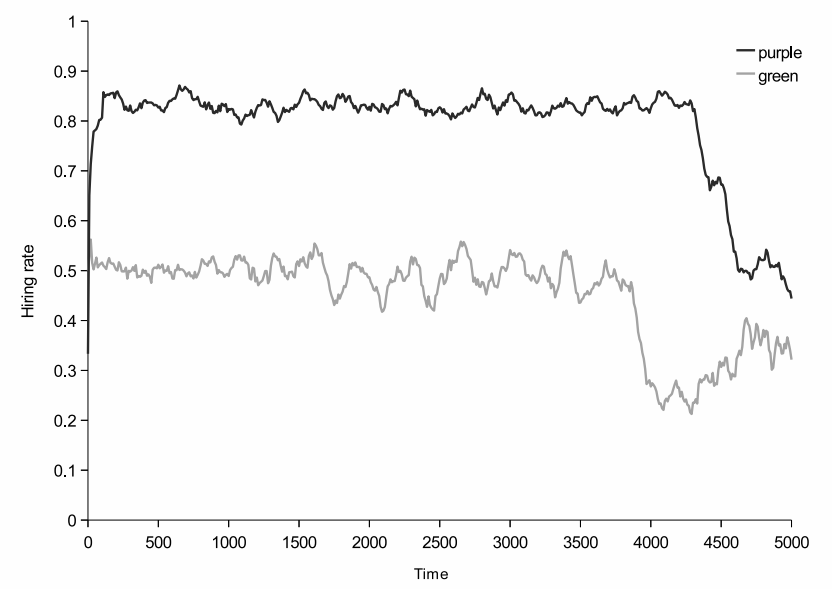


Figure 3.7: Hiring rates for an example run of model variant II (6 purple workers, 4 green workers). Parameters: $\alpha_{employer} = 0.08$, $\gamma_{employer} = 0.21$ (for all initial mappings), $\alpha_{worker} = 0.01$, $\gamma_{worker} = 0.3$

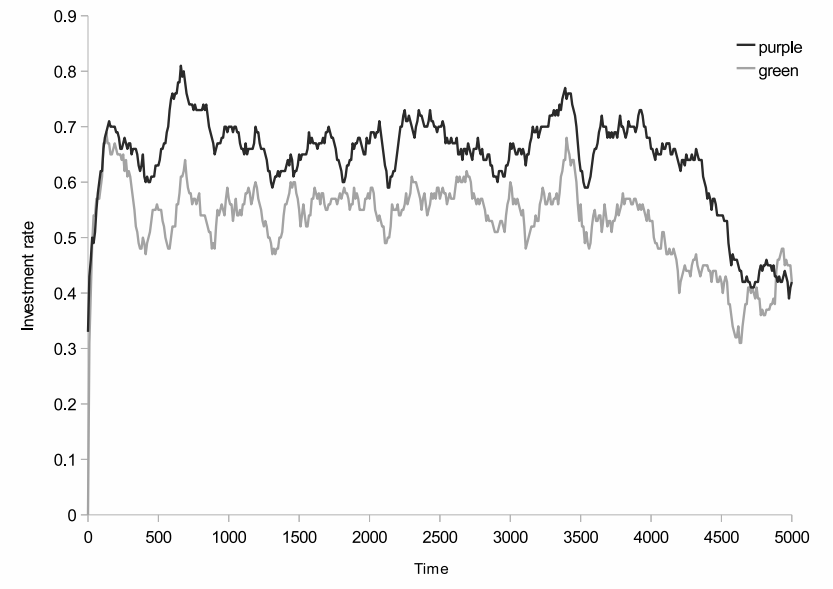


Figure 3.8: Investment rates for an example run of model variant II (6 purple worker, 4 green workers) . Parameters: $\alpha_{employer} = 0.08$, $\gamma_{employer} = 0.21$ (for all initial mappings), $\alpha_{worker} = 0.01$, $\gamma_{worker} = 0.3$

0.4 for green workers. The actual average hiring rates are 0.79 for purple and 0.45 for green workers. This means that employers hire purple workers most of the time even if the test result is bad, whereas the test outcome plays a more important role for green workers (even though they are hired more frequently than even the liberal strategy would suggest). Which beliefs exactly form during simulations with such outcomes is shown in more detail in section 3.5.2.

3.5.1.4 Variant III

Figure 3.4 illustrates that there is no parameter setting supporting discrimination. At most, if at all, it seems that large choice parameters ($\alpha \approx 0.1$) on the employer side, together with small choice parameters ($\alpha \approx 0.05$) on the worker side generate differences in hiring levels.

The simulation with the largest average difference in hiring rates emphasises this result (Figures 3.9 and 3.10). Hiring rates in general are much lower (on average 0.31 for purple and 0.23 for green agents) as in the previous examples. It is difficult for employer agents to distinguish between the benefits of a good and a bad test result, so that they tend to rather not hire anybody. Although there are differences in hiring rates across both groups, this variation does not follow a stable pattern; the green workers simply experience more erratic phases of hiring and investment. This does lead to short cycles of discrimination (e.g. between steps 2000 and 3000), but the pattern does not persist.

3.5.2 Average Results

After finding out appropriate learning parameters, this section presents simulations with more agents, samples of different θ values and more runs,

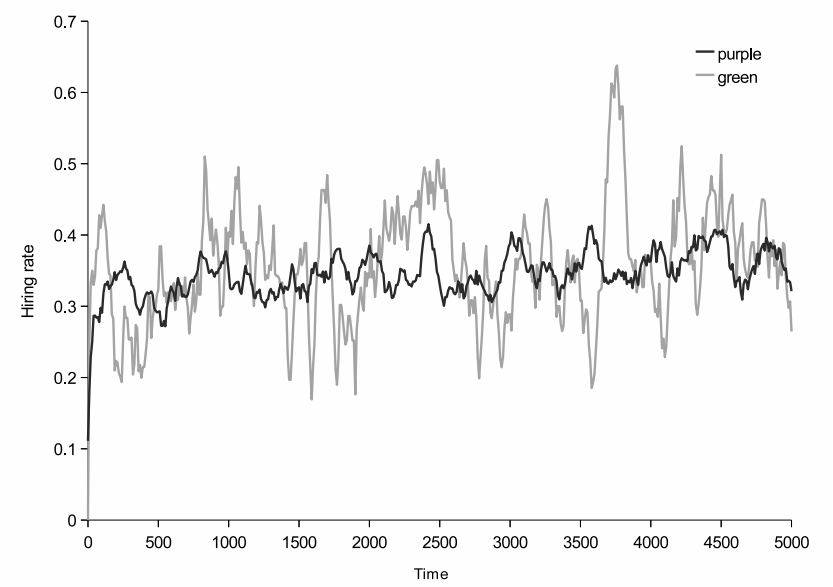


Figure 3.9: Hiring rates for an example run of model variant III (4 purple workers, 6 green workers). Parameters: $\alpha_{employer} = 0.05$, $\gamma_{employer} = 0.07$, $\alpha_{worker} = 0.12$, $\gamma_{worker} = 0.15$

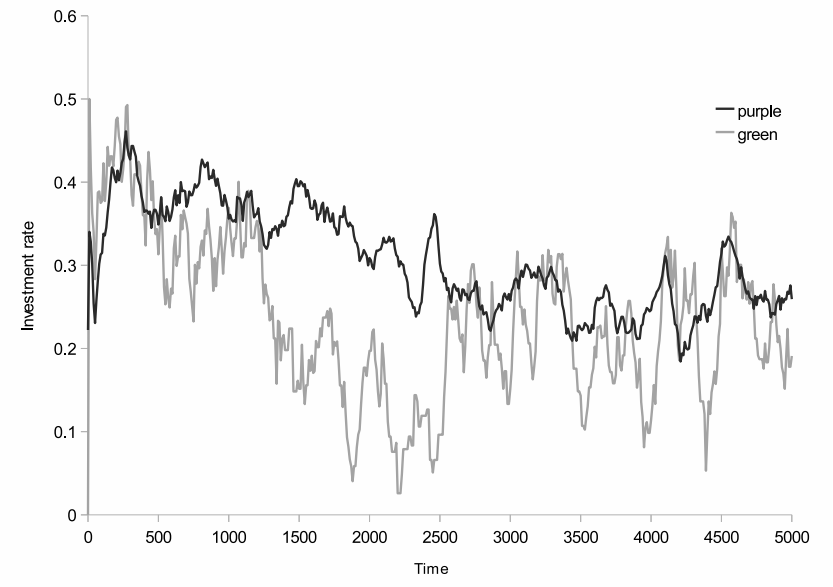


Figure 3.10: Investment rates for an example run of model variant III (4 purple workers, 6 green workers). Parameters: $\alpha_{employer} = 0.05$, $\gamma_{employer} = 0.07$, $\alpha_{worker} = 0.12$, $\gamma_{worker} = 0.15$

keeping the RL parameters constant.

The distribution of θ determines the outcome of the test result, depending on whether an agent has invested or not. If $f_u(\theta)$ and $f_q(\theta)$ is similar, investing does not make a big difference - the outcome is mostly random. If the difference $\delta_{f(\theta)} = f_q(\theta) - f_u(\theta)$ is large ($f_q(\theta) > f_u(\theta)$; see also section 3.2), investing increases the chance of a good test result strongly. If there is a positive relationship between θ and investment level and, consequently, in the hiring rate in the RL model, then the employment level can be expected to increase with $\delta_{f(\theta)}$. The medium range of $\delta_{f(\theta)}$ reflects a similar setup as in the previous section. The chance of a positive test result is slightly higher for investing workers. This setting can also be expected support discrimination the most: It depends crucially on employers' response whether investment pays. If the standards are very high (e.g., never hire a worker with a bad test result if coming from a certain group), investing becomes a costly choice for workers. Conversely, it becomes expensive to turn around such beliefs once they exist for the same reason. In the RL model, this could turn into reinforcing the non-investment choice.

To fix α and γ, the averages of simulations producing an average discrimination rate of 10% were selected, ensuring the possibility of discrimination in subsequent runs. This boundary is set arbitrary to pick not only one, possibly unrepresentative, parameter value of, e.g., the most discriminatory outcome. Furthermore, only model variants I and II are followed up, since variant III was not capable of producing discrimination. The following section will analyse both models further.

Simulations are run for 50 samples of θ, drawn from uniform distributions $f_q\theta$ and $f_u\theta$. Each simulation is repeated 5 times. Table 3.6 summarises the parameters.

Parameter	value	meaning
Discrimination parameters		
$f_q\theta$	0.5 - 1.0	Probability of good test result if invested
$f_u\theta$	0.0 - 0.5	Probability of good test result if not invested
c	0 - 0.1	Investment cost interval
Variant I		
$\alpha_{employer,r^1}$	0.13	learning parameter for r^1 - action set bound to descriptor $\mathcal{L}_0^1$ (test-result is ambiguous and (colour is green or colour is purple))
$\gamma_{employer,r^1}$	0.15	discount parameter for r^1
α_{worker}	0.0107	learning parameter for worker rule
γ_{worker}	0.16	discount parameter for worker rule
Variant II		
$\alpha_{employer,r^1}$	0.05	learning parameter for r^1 - action set bound to descriptor $\mathcal{L}_0^1$ ((test-result=+-) and (colour=green or colour=purple))
$\gamma_{employer,r^1}$	0.43	discount parameter for r^1
$\alpha_{employer,r^2}$	0.05	learning parameter for r^2 - action set bound to descriptor $\mathcal{L}_0^2$ ((test-result=++) and (colour=green or colour =purple))
$\gamma_{employer,r^2}$	0.43	discount parameter for r^2
$\alpha_{employer,r^3}$	0.05	learning parameter for r^3 - action set bound to descriptor $\mathcal{L}_0^3$ ((test-result is bad) and (colour is green or colour is purple))
$\gamma_{employer,r^3}$	0.43	discount parameter for r^3
α_{worker}	0.0107	learning parameter for worker rule
γ_{worker}	0.27	discount parameter for worker rule

Table 3.6: Simulation parameters for obtaining average results.

3.5.2.1 Variant I

Figure 3.11 shows simulation results averaged over different parameter settings θ. On the x-axis the difference $\delta_{f(\theta)} = f_q(\theta) - f_u(\theta)$ is displayed. To compare the employment levels over all the runs, the members of green and purple groups are relabelled into advantaged and disadvantaged depending on which group had the higher or lower employment in a particular simulation. Figure 3.12 displays the results for all simulations.

Figure 3.11 shows that the difference between the groups is small; the largest average difference between employment rates is 0.16. The chance to generate a positive test-result has thus, on average, no influence on discrimination. Moreover, employment does not rise with $\delta_{f(\theta)}$. If discrimination evolves, it is again an isolated event without any relationship to the parameter θ (estimating the linear regression function in the form of $discrimination = \alpha + \beta\delta_{f(\theta)}$ did not result in significant coefficients). This is also underlined by a look at the distribution of single runs in figure 3.12 - runs with the same $\delta_{f(\theta)}$ can result in very different employment levels. Summary measures of the simulation samples given in appendix B illustrate that higher average discrimination is due to single runs with discrimination, whereas most samples show little difference between the groups.

Figures 3.13a to 3.13d show the simulations with the highest difference between employment levels (0.1603 and 0.1604). The pattern is similar to the simulation presented in the previous section 3.5.1.2. Discrimination and investment behaviour evolves at the same time early in the simulation and persists.

Tables 3.7 and 3.8 show the rules that emerged. Since only the rule for ambiguous test results was not fixed, there are just three possible outcomes. Using the exponential selection rule (equation 2.4), the table displays the

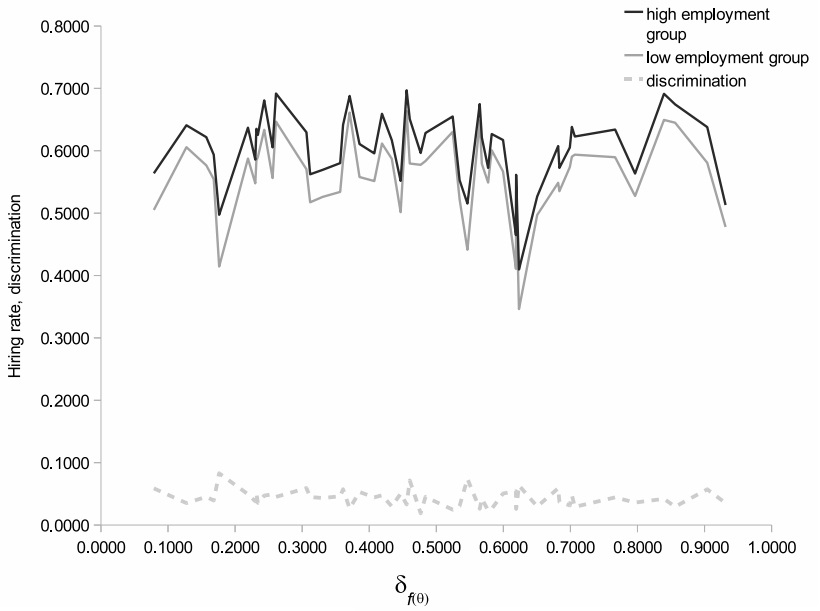


Figure 3.11: Average hiring rates for model variant I across different values of θ. The x-axis is given by $\delta_{f(\theta)} = f_q(\theta) - f_u(\theta)$. Discrimination is the difference between the high and low employment group.

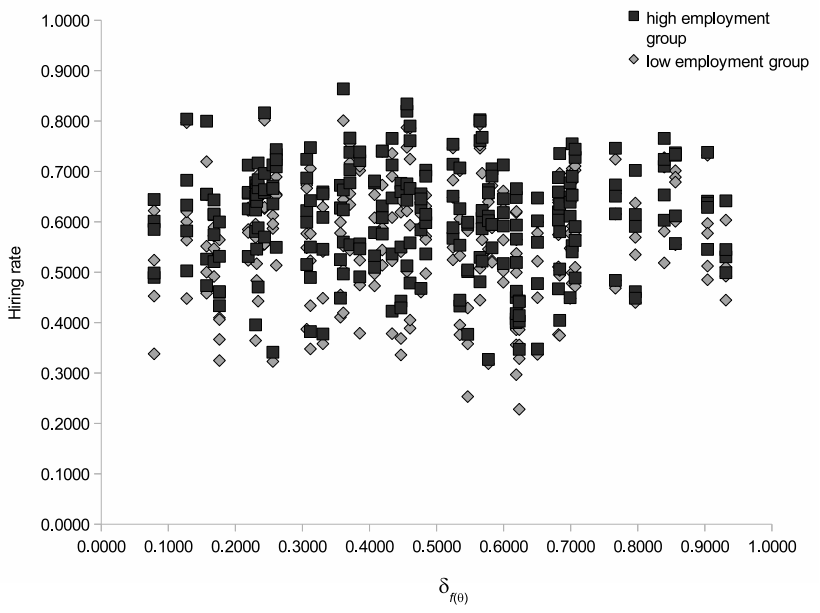
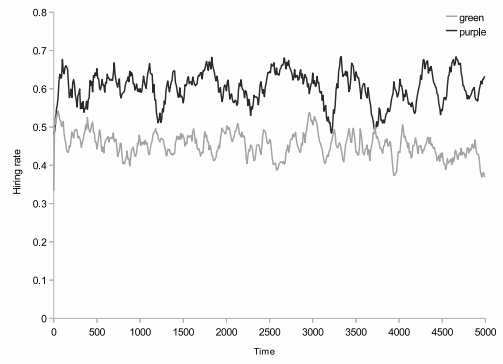
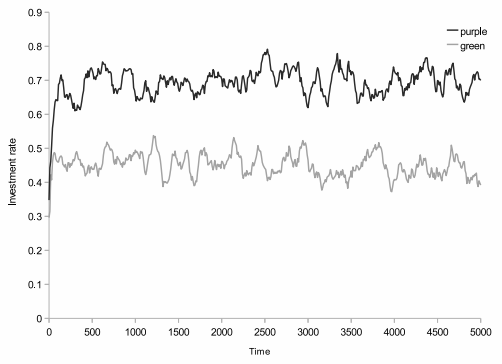


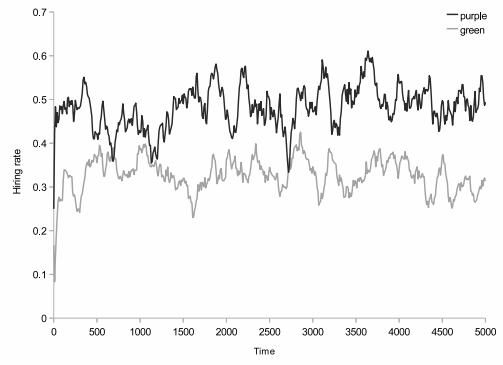
Figure 3.12: Hiring rates for model variant I across different values of θ. The x-axis is given by $\delta_{f(\theta)} = f_q(\theta) - f_u(\theta)$



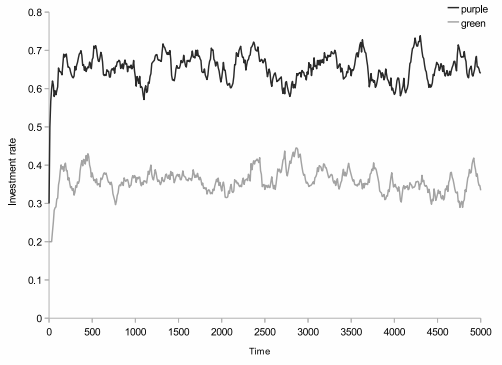
(a) Hiring rates variant I (23 purple, 27 green workers), $f_u(\theta) = 0.26$, $f_q(\theta) = 0.59$



(b) Investment rates variant I (23 purple, 27 green workers), $f_u(\theta) = 0.26$, $f_q(\theta) = 0.59$



(c) Hiring rates variant I (30 purple, 20 green workers), $f_u(\theta) = 0.48$, $f_q(\theta) = 0.55$



(d) Investment rates variant II (30 purple, 20 green workers), $f_u(\theta) = 0.48$, $f_q(\theta) = 0.55$

Figure 3.13: Statistical discrimination - 2 sample simulation runs of model variant I, $n_{employer} = 25, n_{worker} = 50$. Each graph shows moving averages over 10 time steps.

choice propensity $p(hire)$ for action 'hire' and the number of rule activations. The propensity for action 'not hire' is $1 - p(hire)$. The table shows that the original rule (test result is ambiguous and (colour = green colour = purple)) is activated only a few times, while the successor rules are activated more often. Thus, most of the time the discriminatory behaviour persists. However, the differences between the two groups is small. In the first simulation, for example, purple workers are hired with a probability of 0.86, whereas green workers with probability 0.61. This reflects the observation made in section 3.5.1.2 - discrimination comes about due to different investment behaviours alone.

state description	p(hire)	rel. act.	abs. act.
Test-result is ambiguous and (colour = purple or colour = green)	0.61	0.03	3
Test-result is ambiguous and colour = green	0.61	0.47	45
Test-result is ambiguous and colour = purple	0.78	0.49	47

Table 3.7: Rules generated in a sample simulation of model variant I (23 purple workers, 27 green workers), $f_u(\theta) = 0.26$, $f_q(\theta) = 0.59$ and their relative (rel. act.) and absolute (abs. act.) activation frequency in the employer population. Measurements were taken every 100 time steps.

3.5.2.2 Variant II

Figure 3.14 shows again simulation results averaged over different parameter settings θ. Figure 3.15 displays the results for all simulations.

Discrimination is on average higher as compared to setup I. The difference between high and low employment groups moves up to about 0.3. Estimating a linear regression of the form $discrimination = \alpha + \beta\delta_{f(\theta)}$ results in a small, however significant relationship with $discrimination =$

state description	p(hire)	rel. act.	abs. act.
Test-result is ambiguous and (colour = purple or colour = green)	0.67	0.06	19
Test-result is ambiguous and colour = green	0.68	0.49	152
Test-result is ambiguous and colour = purple	0.86	0.45	141

Table 3.8: Rules generated in a sample simulation of model variant I (30 purple workers, 20 green workers), $f_u(\theta) = 0.48$, $f_q(\theta) = 0.55$ and their relative (rel. act.) and absolute (abs. act.) activation frequency in the employer population. Measurements were taken every 100 time steps.

$0.03467 + 0.03617\delta_{f(\theta)}$. However, large differences in average employment levels are again mainly due to extreme values in the samples, as shown in appendix B. A large average discrimination comes usually with a high standard deviation. Figure 3.15 illustrates this graphically. Thus, also here one cannot assume a relationship between θ and discrimination.

Figures 3.16a to 3.16d show the two single simulation runs with the highest discrimination (0.31 and 0.28). In both examples, investment levels are relatively stable from the beginning, while employment levels adjust only after some 1000 time steps. This points to a pattern where first some actual difference between worker group behaviour exists, which is then followed by an adjustment of the beliefs on the employer side.

Tables 3.9 and 3.10 show the rules that emerged for the two sample simulations. In the first simulation, employers developed no rule for the ambiguous test case. They trust most of the time that green workers' good test results lead to high productivity, while they believe the opposite of purple workers. If the test result is bad or ambiguous employers tend not to hire. In the second simulation - reflecting the Fryer results - employers

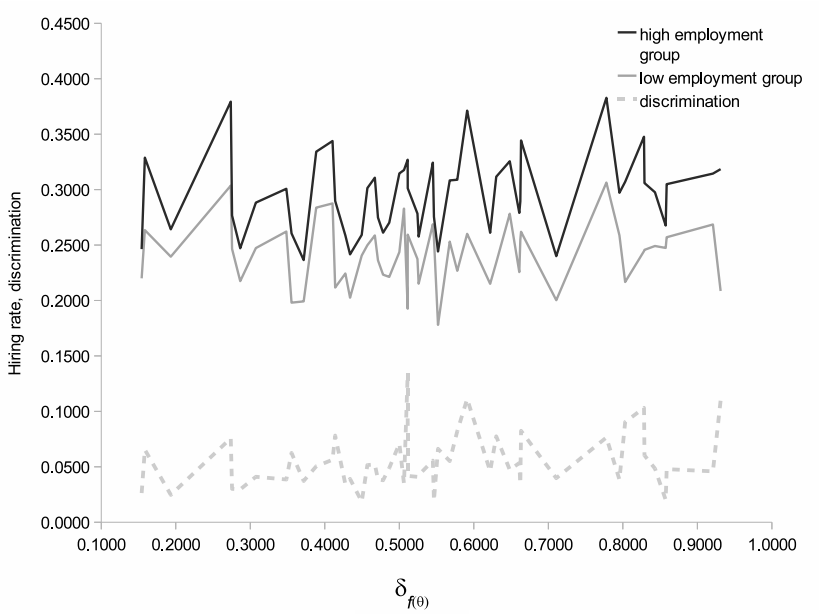


Figure 3.14: Average hiring rates for model variant II across different values of θ. The x-axis is given by $\delta_{f(\theta)} = f_q(\theta) - f_u(\theta)$. Discrimination is the difference between high and low employment group.

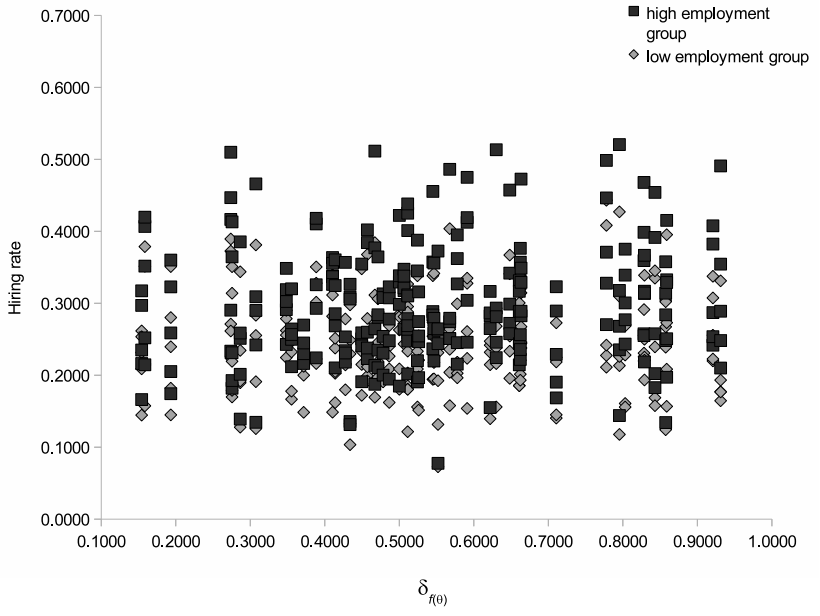
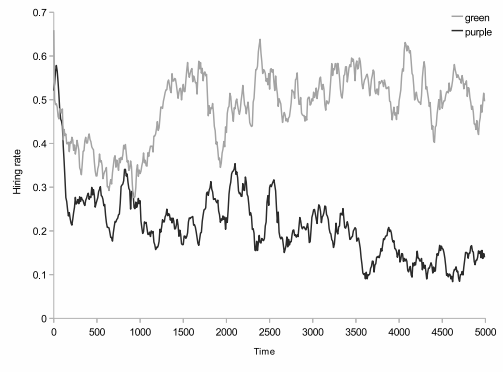
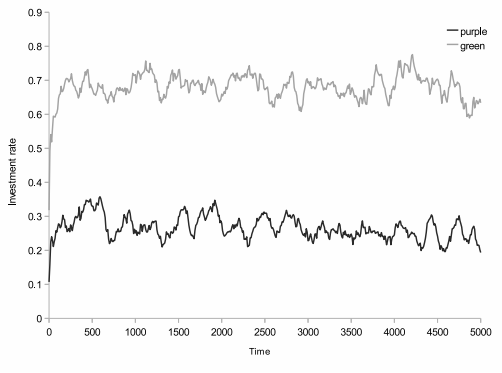


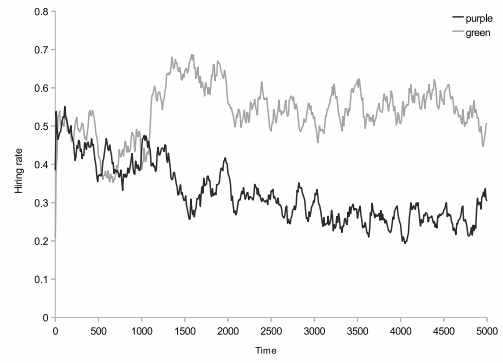
Figure 3.15: Average hiring rates for model variant II across different values of θ. The x-axis is given by $\delta_{f(\theta)} = f_q(\theta) - f_u(\theta)$



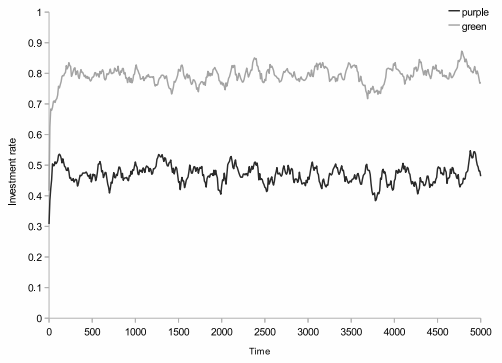
(a) Hiring rates variant II (28 purple, 22 green workers), $f_u(\theta) = 0.01$, $f_q(\theta) = 0.99$



(b) Investment rates variant II (28 purple, 22 green workers), $f_u(\theta) = 0.01$, $f_q(\theta) = 0.99$



(c) Hiring rates variant II (26 purple, 24 green workers), $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$



(d) Investment rates variant II (26 purple, 24 green workers), $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$

Figure 3.16: Statistical discrimination - 2 sample simulations runs of model variant II, $n_{employer} = 25, n_{worker} = 50$. Each graph shows moving averages over 10 time steps.

tend to favour the green workers even in the event of a bad test result; they behave similarly if the result is unclear. Purple workers are always believed to be less productive: If their test result is negative, they get almost never hired; if the result is positive, they are hired only with a chance of 0.2. Likewise, their chances to get hired in case of an ambiguous test result are worse.

The difference between the two samples is the impact of $\delta_{f(\theta)}$. While in the first simulation, employers can be certain that an investing worker has a positive test result ($f_q(\theta) = 0.99$) and a non-investing worker most likely has a negative result ($f_q(\theta = 0.06)$), this is not so clear in the second simulation. In the latter, the chance of a good test result if not investing is closer to the chance of a good result if investing. Consequently, the variety of rules emerging is greater: In the first simulation, the parent state-descriptions 'test-result is good and (colour=green or colour=purple)' and 'test-result is bad and (colour=green or colour=purple)' are activated almost as many times as their children, indicating that the coarser grained descriptions are (on average) nearly as good as the more detailed successors. The expected value of the parent approaches the payoffs in table 3.1. Since the test-result is a certain indicator of productivity, there is no need to consider colour as a hint. In the second simulation, the difference between the expected values of the parent state-descriptions cannot be so large as in table 3.1, because non-investing workers of the same colour will more often get a positive test result. So it becomes more likely that the algorithm evolves (or switches between) more branches, using colour as an additional hint. As the new rules match worker behaviour, they remain stable. As a result, the employers in model variant II follow clearly a discriminatory pattern that makes it difficult for purple workers to escape their situation - even if they achieve good test results, employers are unlikely to believe them.

state description	p(hire)	rel. act.	abs. act.
Test-result is ambiguous and (colour = purple or colour = green)	0.05	0.02	90
Test-result is bad and (colour = purple or colour = green)	0.16	0.18	1040
Test-result is good and (colour = purple or colour = green)	0.33	0.12	698
Test-result is bad and colour = green	0.1	0.2	1152
Test-result is bad and colour = purple	0.05	0.24	1387
Test-result is good and colour = green	0.9	0.12	691
Test-result is good and colour = purple	0.11	0.11	607

Table 3.9: Rules generated in a sample simulation run of model variant II (28 purple workers, 22 green workers), $f_u(\theta) = 0.06$, $f_q(\theta) = 0.99$ and their relative (rel. act) and absolute (abs. act.) activation frequency in the employer population. Measurements were taken every 100 time steps.

3.5.3 How Persistent is Discrimination?

So far, the simulations showed that discrimination in the RL model can emerge. However, there is no general rule when this might happen. Furthermore, looking at the details of variant I, clearly this candidate does not match the empirical results of Fryer Jr. et al (2005). Variant II has more parallels in aggregate results as well as in the behaviour patterns that emerge. In what follows, model variant I is, therefore, not considered any further.

The purpose of this section is to find out whether there are conditions that support statistical discrimination on the average, that is, whether it is possible to make some general statements about why and when discrimination emerges in the RL model. For example, the existence of negative stereotypes towards one worker group could discourage this group from investing from the beginning and persist over time. Such scenarios can be

state description	p(hire)	rel. act.	abs. act.
Test-result is ambiguous and (colour = purple or colour = green)	0.31	0.07	447
Test-result is bad and (colour = purple or colour = green)	0.1	0.06	794
Test-result is good and (colour = purple or colour = green)	0.35	0.13	365
Test-result is ambiguous and colour = green	0.63	0.08	482
Test-result is ambiguous and colour = purple	0.33	0.07	447
Test-result is bad and colour = green	0.18	0.19	1170
Test-result is bad and colour = purple	0.06	0.2	1256
Test-result is good and colour = green	0.86	0.11	662
Test-result is good and colour = purple	0.2	0.09	591

Table 3.10: Rules generated in a sample simulation run of model variant II (26 purple workers, 24 green workers), $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$ and their relative (rel.act.) and absolute activation (abs. act.) frequency in the employer population. Measurements were taken every 100 time steps.

modelled by starting with situations in which discrimination exists, for example, negative stereotypes or uneven cost distributions. Then, it can be observed in which direction the simulation develops further.

Three scenarios are considered to investigate this question. First, taste-based discrimination is introduced. In this scenario, the share of firms never hiring green workers is increased. In the second scenario, heterogeneous conditions for green workers are introduced by increasing their investment cost for an initial, but limited period. In the third scenario, employers are confronted with always investing purple and never investing green workers for an initial, limited period. After this period, the deterministic workers are replaced with the original, homogenous agents. The third scenario can also be thought of as an extreme case of the second where investment cost

at the beginning is prohibitive for green workers, and 0 for purple workers.

Taste-based discrimination In this scenario, inequality is generated by fixing firm behaviour, similar as the preference model of Becker (1957)). For this purpose, simulations are run with a proportion of firms never hiring green workers; for purple workers, the same rules as in variant II apply. Figures 3.17 and 3.18 shows the hiring rates of green and purple workers as the number of these firms (labelled p-firms) increases. In this scenario, the employment chances of green workers worsen deterministically. The question is how they react to these conditions and how this influences the remaining firms hiring green workers. If statistical discrimination is encouraged, one would expect an over-proportional decrease in hiring levels of green workers: Green workers invest less due to worsening conditions on the labour market, which induces the remaining liberal employers not to hire them because of expected lower investments. If the liberal employers continue to hire greens at the same rate, then worker behaviour reflects just the increasing number of p-firms.

Figure 3.17 shows a slight sigmoid shape of the hiring level graph of green workers, that is, hiring levels decrease slightly over-proportionally while the number of p-firms increases linearly. An estimate of the logit function with employment as dependent and share of p-firms as independent variable (interpreting the employment levels as categories) shows graphically a closer approximation than the linear model (coefficient estimates are significant). That is, in the medium region workers are discouraged strongly from investing, resulting in over-proportionally lower hiring levels. However, the effect is small.

Furthermore, as figure 3.18 shows, the discrimination of the green group has also an effect on the hiring level of purple workers. The effect is linear.

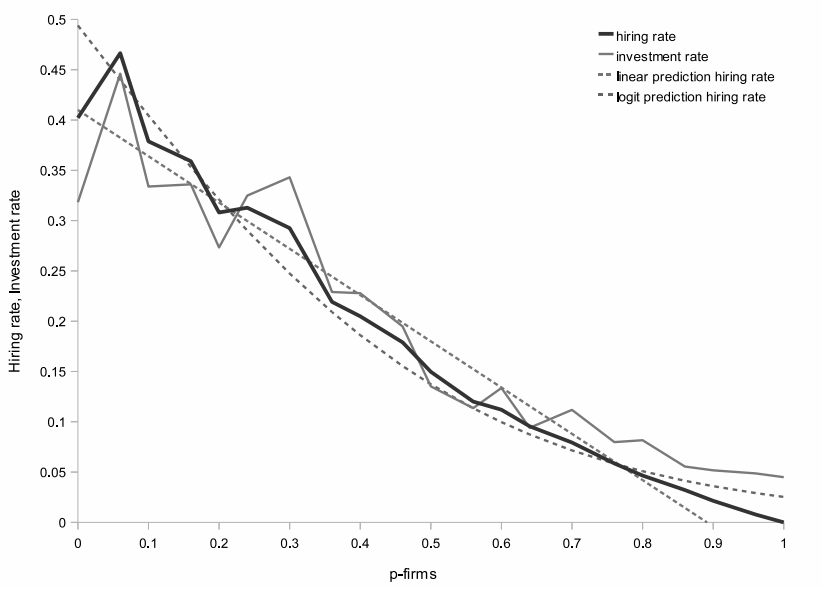


Figure 3.17: Hiring rates of green workers if increasing numbers of firms deterministically discriminate against them. The x-axis depicts the number of firms that never hire green workers. The dotted line indicates the linear model estimated from the data, the hatched line the estimated logit function.

In the beginning, purple workers manage to free-ride on the expectations of the employer population and invest at lower rates as they get hired. Firms expect higher average investment rates (independent of which colour invests more), which makes riskier firm decisions, such as hiring workers with a bad test result, more profitable. With increasing p-firms, the chance of generating high payoffs on average decreases as green workers invest less and less, so the average payoff of hiring (any) workers with a bad or ambiguous test result will also decrease.

Unequal investment costs In this scenario, there is initially an unequal distribution of costs similar as described above in the experiment of Fryer Jr. et al (2005). For a starting period, the cost distribution of green workers is drawn from the higher interval 0.1 - 0.3, so that investing is always more expensive than for the purple group (interval 0 - 0.1). This could repre-

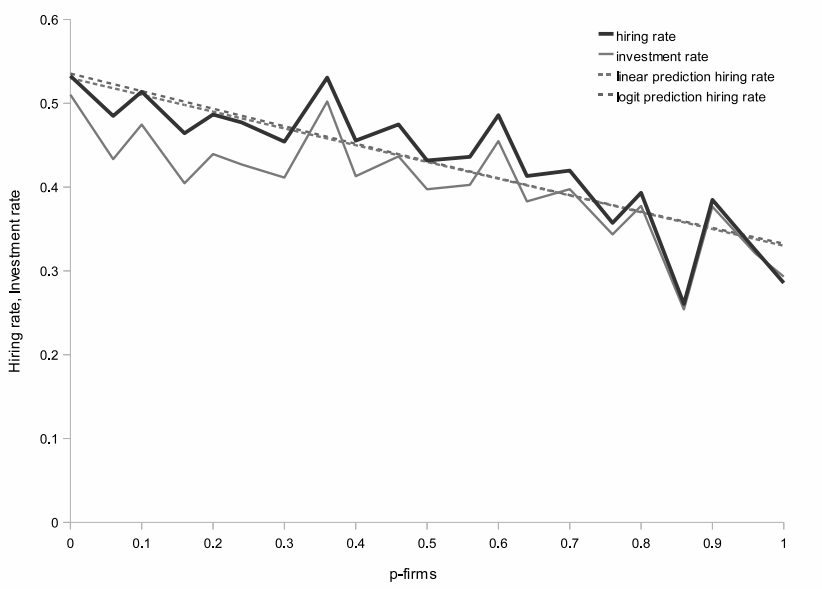
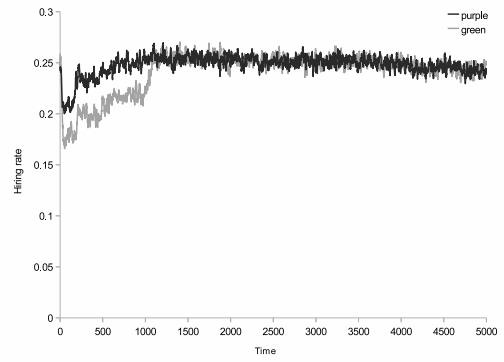


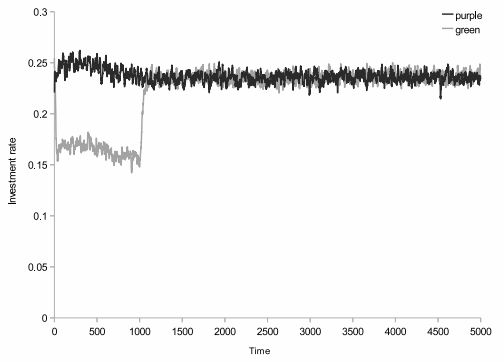
Figure 3.18: Hiring rates of purple workers if increasing numbers of firms deterministically discriminate against green workers. The x-axis depicts the number of firms that never hire green workers. The dotted line indicates the linear model estimated from the data, the hatched line the estimated logit function.

sent a situation where entry barriers into certain vacations are high for the discriminated group. The question is whether this leads to different investment behaviour and if yes, whether this persists after the barrier is removed. The scenario implemented by setting back the cost distribution to normal after step 1000. Simulations are run again for $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$ and $f_u(\theta) = 0.06$, $f_q(\theta) = 0.99$; that is, with the same settings as for the two sample simulations with the highest discrimination from the preceding section 3.5.2. Figures 3.19a to 3.19d show the results.

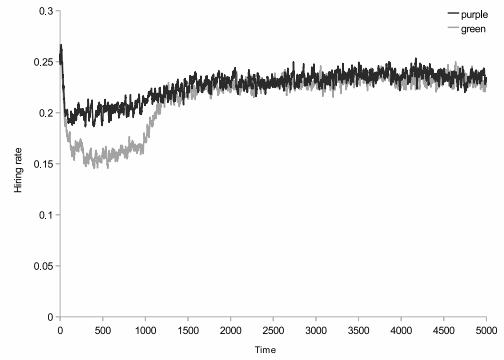
In both scenarios, green workers invest less up to time step 1000, and employers hire them at a corresponding lower rate. The hiring rate differs according to the distributions of θ. In the simulation with less noise ($f_u(\theta) = 0.06$, $f_q(\theta) = 0.99$), the employment level of both groups is higher. The pattern in both simulations is similar. In both simulations, green work-



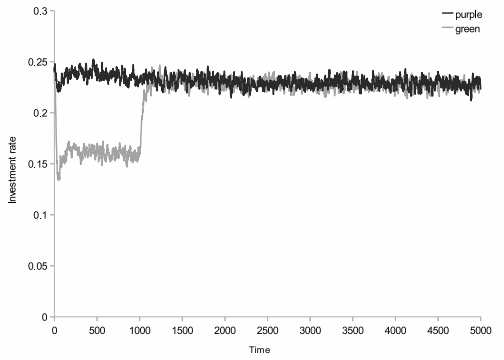
(a) Hiring rates variant II), $f_u(\theta) = 0.01$, $f_q(\theta) = 0.99$



(b) Investment rates variant II, $f_u(\theta) = 0.01$,$f_q(\theta) = 0.99$



(c) Hiring rates variant II, $f_u(\theta) = 0.35$,$f_q(\theta) = 0.98$

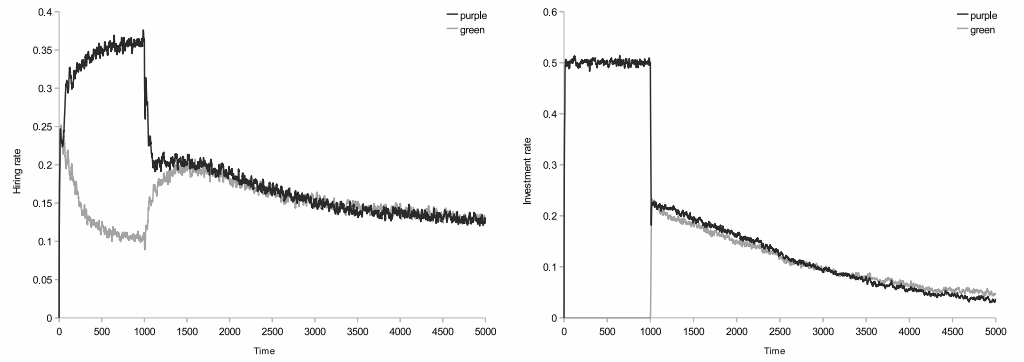


(d) Investment rates variant II, $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$

Figure 3.19: Effect of cost heterogeneity in model variant II, $n_{employer} = 25, n_{worker} = 50$, 25 repetitions. Each graph shows moving averages over 10 time steps.

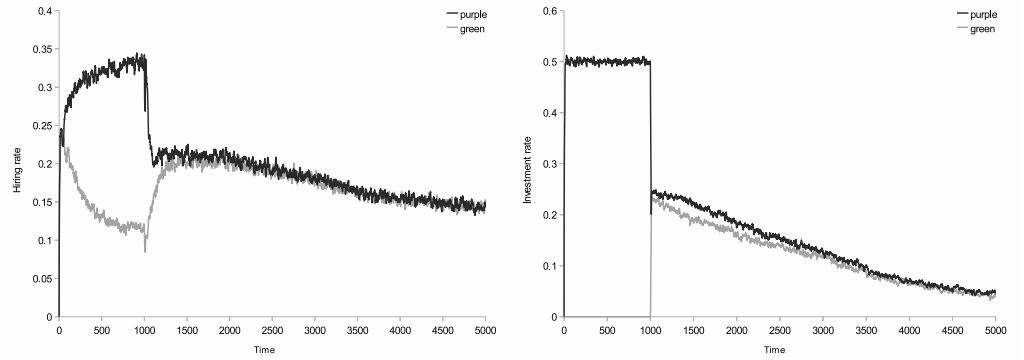
ers are hired at a similar rate as purple workers after the barrier is removed. That is, cost heterogeneity leads to discrimination, but the effect on employer beliefs is not permanent. Whether this is because the difference is too small can be checked in the next paragraph.

Negative stereotypes In the last scenario, inequality is generated by creating negative stereotypes on the employer side. To achieve this situation, the two sample simulations of variant II are set up with the same parameters as before. The simulation is split in two parts: For 1000 steps,



(a) Hiring rates variant II, $f_u(\theta) = 0.06$, $f_q(\theta) = 0.99$

(b) Investment rates variant II, $f_u(\theta) = 0.01$,$f_q(\theta) = 0.99$



(c) Hiring rates variant II, $f_u(\theta) = 0.35$,$f_q(\theta) = 0.98$

(d) Investment rates variant II, $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$

Figure 3.20: Effect of prior negative stereotypes in model variant II, $n_{employer} = 25, n_{worker} = 50$, 25 repetitions. Each graph shows moving averages over 10 steps.

employers are confronted with purple workers who always, and green workers who never invest. After that, all deterministic worker agents are removed and replaced by learning worker agents as in the original setup. The simulation is then run for another 4000 time steps. Figures 3.20a to 3.20d shows average results for 25 repetitions.

As the figures illustrate, employers discriminate on average when worker behaviour is deterministic. They hire purple workers at a rate of almost 0.35 and green workers at a rate of about 0.1. However, after exchanging the worker agents, both hiring rates converge to the same rate in between the

extremes. The only difference between the two samples is the employment level: For the setting $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$, the level is higher for purple and lower for green workers as compared to the first simulation. Furthermore, there is a slightly lower investment. So it seems that the smaller chance of getting a positive test result discourages green workers from investing. This effect is small and only temporary. In the longer run, both hiring and investment rates converge.

Tables 3.11 and 3.12 show the rules responsible for this result. In the first simulation ($f_u(\theta) = 0.06$, $f_q(\theta) = 0.99$), the relative frequency of activations of the general rule 'if test-result is bad and (colour=green or colour=purple)' increased from 0.13 to 0.24, whereas the share of its children decreased. Thus, after switching worker behaviour, employers generalised some rules. For simulation $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$, the share of the general rule 'if test-result is good and (colour=green or colour=purple)' increased only slightly from 0.1 to 0.15. Thus, employers generalise existing discriminating rules to some extent. However, the adaptation process works mainly over adjusting the selection probabilities.

Some more simulations were run to verify the observation that initial beliefs do not influence the result in the longer run. Figure 3.21 shows the discrimination between green and purple workers for different $\delta_{f(\theta)}$. On average, discrimination is low; maximum values are at most around 0.1. Averaged over all steps, there was no simulation with discrimination larger than 0.06. The extent of discrimination varies; this variation, however, does not occur between simulations, but over time. In most simulations, green workers even get hired more often at some stage. For example, for $f(\theta_q) = 0.71$ and $f(\theta_u) = 0.21$ discrimination is close to -0.15 at $t = 1500$, but close to 0.05 at $t = 2000$ and $t = 4500$.

state description	p(hire)	rel. act.	abs. act.
for $t < 1000$			
Test-result is ambiguous and (colour = purple or colour = green)	0.41	0.02	20
Test-result is ambiguous and colour = green	0.5	0.01	11
Test-result is ambiguous and colour = purple	0.5	0.01	11
Test-result is bad and (colour = purple or colour = green)	0.33	0.13	119
Test-result is bad and colour = green	0.13	0.21	185
Test-result is bad and colour = purple	0.84	0.2	184
Test-result is good and (colour = purple or colour = green)	0.34	0.15	131
Test-result is good and colour = green	0.19	0.13	118
Test-result is good and colour = purple	0.78	0.13	119
for $t = 1000$ to $t = 5000$			
Test-result is ambiguous and (colour = purple or colour = green)	0.15	0.02	136
Test-result is ambiguous and colour = green	0.45	0.02	121
Test-result is ambiguous and colour = purple	0.54	0.02	124
Test-result is bad and (colour = purple or colour = green)	0.39	0.24	1791
Test-result is bad and colour = green	0.13	0.18	1353
Test-result is bad and colour = purple	0.14	0.18	1375
Test-result is good and (colour = purple or colour = green)	0.37	0.11	815
Test-result is good and colour = green	0.16	0.12	866
Test-result is good and colour = purple	0.16	0.12	877

Table 3.11: Rules generated for model variant II with negative stereotypes, 25 repetitions, $f_u(\theta) = 0.06$, $f_q(\theta) = 0.99$, and their relative (rel. act.) and absolute (abs. act.) activation frequency in the employer population before and after time=1000. Measurements were taken every 100 time steps.

state description	p(hire)	rel. act.	abs. act.
for $t < 1000$			
Test-result is ambiguous and (colour = purple or colour = green)	0.35	0.08	74
Test-result is ambiguous and colour = green	0.34	0.07	61
Test-result is ambiguous and colour = purple	0.63	0.07	63
Test-result is bad and (colour = purple or colour = green)	0.37	0.13	121
Test-result is bad and colour = green	0.25	0.1	93
Test-result is bad and colour = purple	0.7	0.1	94
Test-result is good and (colour = purple or colour = green)	0.36	0.1	93
Test-result is good and colour = green	0.16	0.17	160
Test-result is good and colour = purple	0.8	0.17	160
for $t = 1000$ to $t = 5000$			
Test-result is ambiguous and (colour = purple or colour = green)	0.31	0.07	613
Test-result is ambiguous and colour = green	0.32	0.06	553
Test-result is ambiguous and colour = purple	0.32	0.06	560
Test-result is bad and (colour = purple or colour = green)	0.4	0.12	1075
Test-result is bad and colour = green	0.28	0.1	864
Test-result is bad and colour = purple	0.3	0.1	855
Test-result is good and (colour = purple or colour = green)	0.44	0.15	1271
Test-result is good and colour = green	0.17	0.17	1430
Test-result is good and colour = purple	0.19	0.16	1419

Table 3.12: Rules generated for model variant II with negative stereotypes, 25 repetitions, $f_u(\theta) = 0.35$, $f_q(\theta) = 0.98$, and their relative (rel. act.) and absolute (abs. act.) activation frequency in the employer population before and after time=1000. Measurements were taken every 100 time steps.

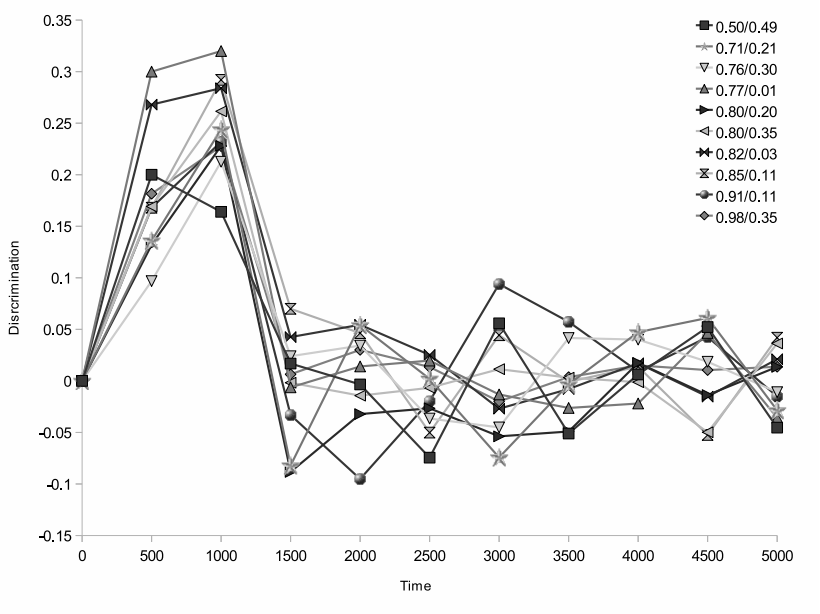


Figure 3.21: Discrimination rates of green workers if firms are biased negatively against green workers. Discrimination is the difference between green and purple employment levels. Each line represents averages of 10 simulation runs for a particular combination of $f_q(\theta)$ / $f_u(\theta)$ (The legend displays the concrete realisations).

3.5.4 Summary of the Simulation Results

Besides presenting more detailed figures about the behaviour of the RL model, the purpose of the preceding two sections was to test under which conditions it is possible to generate discrimination. Varying the exogenous parameter θ produced similar results as already obtained in the exploration section. Discrimination can occur, but cannot be explained by θ alone. In the simulations that resulted in discrimination, a certain independence of worker and employer behaviour was observed. Thus, an important reason for discrimination in the RL model can be seen in different investment levels on the worker side, if they are discovered only later during the simulation by the employers.

In the next step, three scenarios where simulated. These scenarios intro-

duced systematic biases in the simulation setup in the form of deterministically discriminating firms, different investment cost distributions, and the introduction of negative beliefs about green workers on the employer side. The simulation results show a slight tendency of mutually stabilising expectations. So, for example, there is a non-linear relationship between the number of deterministically discriminating firms and investment behaviour. The more firms discriminate one group, the stronger this group is discouraged from investing, and the more unbiased firms tend not to hire members of that group. The next scenario showed that the effect of increasing the investment costs of green workers was not significant enough to establish a persistent negative employer bias. A similar picture exists if firms have negative stereotypes about green workers. Thus, even an initial prohibitive entry cost did not lead to persistent discrimination of the disadvantaged group. In the longer run, discrimination disappears.

Thus, the RL model shows a weak interdependency between employer expectations and worker behaviour. However, the main driving force in the RL model is the stickiness of investment behaviour. This dynamic is closer to the approach of Phelps (1972), where employers adjust to existing differences in worker productivity. In the RL model, this can only occur if the simulation takes a certain path. A favourable condition can be seen in free-riding behaviour of worker agents. If firms treat workers equally in the beginning, but investment behaviour is different, members of the free-riding group have no incentives to increase investments. Nevertheless, as their behaviour is flexible enough, employers will adjust their hiring levels. On the other hand, workers in the model are less flexible, and so they remain with their initial choices. So to speak, initial 'liberal' behaviour in favour of less productive workers can turn into persistent discrimination, but negative beliefs usually do not discourage otherwise homogenous workers

from investing.

3.6 Conclusion

This chapter presented a RL model of statistical discrimination using the BRA algorithm developed in chapter 2. It aimed to reproduce experimental results and asked whether these results could be generalised with an RL-based model. Thus, the question was, whether, starting from empirical observation, there is a general mechanism that could explain the emergence of statistical discrimination.

The RL model was compared with both theoretical and empirical results from statistical discrimination games. Several model variants were simulated to find out which setup and parameter setting can reproduce the patterns of Fryer Jr. et al (2005)'s classroom experiment best. One model (variant II) reproduced with a relative large discrimination the aggregate as well as behavioural patterns. Using this setup, some more scenarios were simulated to analyse the properties of the model further.

Similar to game-theoretic and experimental results, the RL model shows that statistical discrimination can exist. Whether it occurs is, however, path-dependent. The scenarios simulated in this chapter were not capable of creating a setting in which statistical discrimination emerges on the average.

Some differences to the theoretical as well as experimental literature can be highlighted:

- The relationship between θ and employment level could not be observed. Increasing the likelihood of a positive test result does not increase the number of investing workers and thus their hiring rates.

- Initial beliefs do not necessarily influence the outcome in the long run. That is, employer behaviour can adjust very quickly if worker behaviour changes, e.g. due to an intervention such as increasing access to human capital.

- In the RL model, discrimination emerged although no knowledge about market outcomes was available, whereas in the experiment knowledge about market outcomes was public. Thus, for a belief to emerge it may require even less publicly shared information. By looking closer at the rules that emerged during the simulations, it could be shown that the behavioural patterns of the RL model are nevertheless similar as in the classroom experiment: If in doubt, hire a worker if he or she comes from the group that is believed to be more productive; do not hire if he or she comes from the group that is assumed to be less productive. In some simulation runs, this results in a state in which workers of the preferred group get hired even if they signal low productivity.

- In the RL model, worker behaviour is the driving factor for generating discrimination. Discrimination can only emerge if the groups stick to different investment behaviour after employer change their policies.

In summary, the RL model was shown to be a good approximation of actual human behaviour in the experiments. While results of experiments and simulations are similar, the RL model cannot confirm all the relationships postulated by theory. Furthermore, a general rule capable of creating discrimination could not be found.

Chapter 4

Network Formation

4.1 Introduction

Networks are an important paradigm for modelling social and economic relationships. How members of a society are connected to each other determines behaviour and welfare. Through connections to other persons, important resources can be accessed and used for one's own purpose.

A useful distinction is between social and personal networks. Personal networks are comprised of the relations an individual has, e.g. relatives, friendships, acquaintances. Social networks are an aggregation of individual networks.

The structural properties of networks have long been the topic of network analysis in Sociology. The following stylized facts about empirical personal networks can be drawn from this literature:

– Personal networks are small; the closer the contact, the smaller the network. In personal relationships, these contacts are usually family and a few very close friends. Building on existing research, Hamill

and Gilbert (2009) also note that maintaining close relationships is costly, requiring resources as time and effort. This naturally limits the extent of close relationships a person can maintain.

– The distribution of the degree (the number of links a node in the network has) of personal networks is unequal. Some individuals are more sociable, and so have more relationships than most other people in the network. Empirically, distributions of personal networks are typically right-skewed, or 'fat-tailed'. This has been observed, for example, for co-authorship data: There are some economists who appear in many co-authored papers, while the majority has only few co-author relationships (Jackson 2008; p.60).

– Members of personal networks tend to share the same characteristics (homophily). Contacts between similar people are more likely than among dissimilar people. This can be described by the cluster coefficient. This coefficient determines, in principle, the likelihood that two nodes share the same links. Thus, personal networks have characteristically large cluster coefficients, as compared to, say, a social network. This phenomenon has already been observed by Granovetter (1973).

Social networks, on the other hand, are much less connected than individual networks, i.e. they have a low overall network density (the ratio of all links relative to all possible links). In larger groups, it is simply impossible to know most other people. Nevertheless, most individuals in a society can be reached within a few steps. This property of small average path lengths (the number of nodes between any pair of nodes in the network) and small diameters (the largest distance between any two nodes in the network) has been captured in the notion of small worlds. This phenomenon became widely known by Milgram's experiments (e.g. Watts 2004; Milgram and

Travers 1969). In these experiments, persons had to route letters to persons in other states whom they did not directly know, passing the letter to the target themselves or to someone they thought is likely to know that person. About a quarter of the letter reached their targets. Drawing on such insights, Milgram suggested that there are in general six degrees of separation, i.e. anyone in a society is linked to anyone else with just six intermediaries.

While a large literature about the structural properties of networks exists, much less has been written about the dynamic aspects. Many concepts of how and why people relate to each other are based on chance (homophily, social or regional closeness, etc.), but they do not conceptualise the creation and maintenance of relationships as choice. For example, Barabasi's preferential attachment model (Barabasi and Albert 1999) simply assumes a higher probability of linking to already well-connected persons in the society. However, if links, as stated above, are assumed to be costly, persons have to make implicit or explicit decisions about who they want to be friends with.

The concept of strategic network formation models the decisions on the micro-perspective explicitly. Strategy should here be understood not literally, but in the sense that individuals tend to form mutually beneficially relationships and drop relationships that are not (Jackson 2008; p.153).

Viewing the formation of connections in such a way allows to model networks as the outcome of a game. Goyal (2007) summarises the main features of strategic network formation as follows:

– Strategic network formation can be modelled as a game in which players decide to link or not to link to each other, depending on some

value function of the network and an allocation rule that distributes the value among the players.

- It is based on assumptions of complete information (players know each other and the payoff structure).
- Networks have some form of externality; that is, for individual players the structure of the network itself influences their utility.

Communication networks represent a commonly used network model in Economics. Communication networks model relationships among individuals that exhibit some benefit to the members of the network, typically in the form of information flows. The benefit depends on the number of other persons a member is linked to; the more persons in the network, the higher its value to the individual. In its simple form, utility is a linear function of the number of other players in the personal network. A more realistic form assumes decay in value the more distant the source of information is. Establishing and maintaining direct links is costly because it typically involves some effort. Individual utility depends, then, on the relationship between costs and benefits. On the aggregate, then, this relationship determines the shape of the networks that can form. Such a general model can cover many interesting social and economic settings where the structure of the networks influences the well-being of the members, for example, friendships, work relationships, but also research partnerships between firms.

Although strategic network formation focuses on the micro perspective, it is also possible to generate large-scale networks. Jackson and Rogers (2005), for instance, present a spatial variation of a communication network game. In this model, players are distributed on islands. Costs for connecting to near-by players are low and high for connecting to distant

players. Jackson and Rogers (2005) show that with certain cost settings, the resulting network exhibits small world properties. The intuition is that players form links with most of their close neighbours, but economise on distant links. It is, nevertheless, still beneficial to maintain the distant links which provide the only chance to access the benefits of more players. Similarly, the residential segregation model of Schelling (1971) could be seen as a prototype of a network model combining chance and choice: Green and red members of a society move randomly and meet other members. Depending on their preference for living in a same-colour neighbourhood, they decide to relocate or stay. The result is a society that is clustered into same colour neighbourhoods.

In recent years, several experiments with strategic network formation have been conducted in order to compare the theoretic predictions with empirical data. Few of them are based on the partially cooperative network model of Jackson and Wolinsky (1996; JW). In this model, links are formed only if both involved players agree. More research is related Bala and Goyal (2000)'s non-cooperative version, where links can also be established unilaterally.

As this overview illustrates, strategic network formation can be seen as a different and complementary way for generating personal and social networks dynamically. There have also been experiments to evaluate the predictive power of network formation models. So far, however, no experience-based model of network formation exists. In general, RL models have been found to predict experimental data better (see chapter 2). The purpose of this chapter is to provide such a model for network games in order to bridge the gap between theory and experimental evidence. It focuses on the level of personal networks alone.

Section 4.2 first introduces notation and definitions. Section 4.3 shortly discusses the relevant theoretical, section 4.4 the experimental literature. The RL model is then described in section 4.5. The simulations are analysed in section 4.6. In section 4.8, a modified version of the RL model is used to compare the results to the laboratory experiments conducted by Conte et al (2009). The main question is whether the RL model can predict the outcome of network formation processes better than the equilibrium prediction.

Relating the RL network model to the general learning approach as discussed in chapter 2, it represents the case-based variant of BRA. Using definition 4 developed in section 2.4.4, it can be described with: $k > 1$, $|A^k| > 1$, $\bigcap_{k=1}^{n} \mathcal{L}_l^k = \emptyset$ and $succ(\mathcal{L}_0^k) = \emptyset$ for a number k of cases. It is assumed that all players know each other, so that k players represent the k cases. There is no dynamic extraction of rules. A variant of BRA where the case distinctions are allowed to evolve dynamically is shortly presented in section 4.7.

4.2 Definitions and Notation

4.2.1 Graphs

Definition 5. *Graphs. A graph g, $g \subseteq G$, consists of a nonempty set of elements, called vertices and denoted $v_i, v \subseteq V$, and a list of pairs of vertices, called edges. Edges connecting two vertices v_i and v_j directly are denoted ij. A weighted graph is a graph in which weights are attached to the edges. The cardinality of a graph is the number of edges it contains, and is denoted with c_g.*

$\mathcal{N}$ denotes the set of all possible graphs that can be generated from V.

g+ij denotes the graph that can be obtained by adding the edge ij to graph g. Conversely, g-ij denotes the graph obtained by deleting this link.

Graphs that are obtained by adding or deleting links are called 'adjacent'.

For simplifying the description of networks, an undirected, unweighted graph can be defined as follows:

Definition 6. *Network density. Network density measures how strongly the vertices of a graph are interconnected by dividing the number of existing edges by the number of possible edges. In the directed graph it is defined as* $D = \frac{1}{n*(n-1)}\sum_{i=0}^{n}\sum_{j=0}^{n} ij$*, for the undirected graph it simplifies to* $D = \frac{1}{0.5(n*(n-1))}\sum_{i=0}^{n}\sum_{j>i}^{n} ij$*. The fully connected graph has a density of 1, the empty graph a density of 0.*

Definition 7. *Shortest path. Let* P_{xy} *be a nonempty path in a weighted graph g from vertex x to vertex y, consisting of k edges* $xv_1, v_1v_2...v_{k-1}y$*. The weight of* P_{xy}*, denoted as* $W(P_{xy})$*, is the sum of the weights,* $W(xv_1), W(v_1v_2), ...W(v_{k-1}y)$*. If x=y, the empty path is considered to be a path from x to y. The weight of the empty path is zero. If no path between x and y has weight less than* $W(P_{xy})$*, then* P_{xy} *is called a shortest path between x and y, and is denoted as* SP_{xy}*.*

Definition 8. *Average path length. The average path length is the average of all shortest paths in the graph g and denoted as L:* $L = \frac{1}{n}\sum_{i\neq j}^{n} SP_{ij}$

While the above definitions are taken from standard graph theory (e.g. Bondy 2008), the following notation is simply a short way of describing network structures in small networks:

Definition 9. *Network patterns. Let the vector a be the ordered in- or out-degree of all vertices. The in-degree is the number of edges arriving at vertex i,the out-degree is the number leaving from it, the sum of both is called in-out degree. In an undirected graph the in-degree equals the out-degree, since for all edges arriving at i, there must be one leading back. If the labels of the nodes are interchangeable, a describes the structure of the network completely.*

For example, the structure 1,1,1,1,4 represents a star with 5 vertices, four vertices having one link, denoted by '1', and one vertex having four links to all other vertices, denoted by '4'.

4.2.2 Games on Graphs

In a network game, the vertices v_i represent players, and the edges the relationships they can engage in.

Network games further include value and allocation functions on the set of possible graphs G. Value functions specify how the total utility is generated by the network, and the allocation rule defines how this value is distributed among the individual players.

Definition 10. *Value functions (see Jackson and Wolinsky (1996)).*
(i) A value function vf is a mapping $vf : \{g|g \subset g^{\mathcal{N}}\} \to \mathbb{R}$
(ii) The value function cvf *is defined as the sum of individual utilities of the players:* $cvf(g) = \sum_i u_i(g)$

Definition 11. *Allocation function (see Jackson and Wolinsky (1996)).*
An allocation function $Y : \{g \mid g \subset g^{\mathcal{N}}\}$ *distributes the value generated by* vf*. The 'equal split rule' (Jackson and Wolinsky 1996) distributes the value evenly among the players and is defined as:* $Y_e(g, v) = cvf(g)/n$.

4.2.3 Stability definitions

Definition 12. *Pairwise Stability (Jackson and Wolinsky 1996). A network is pairwise stable if*
(i) for all edges $ij \in g$, $Y_i(g, v) \geq Y_i(g - ij, v)$ *and* $Y_j(g, v) \geq Y_i(g - ij, v)$
(ii) for all edges $ij \notin g$, $Y_i(g, v) < Y_i(g + ij, v)$ *then* $Y_j(g, v) > Y_i(g + ij, v)$

In words: If a link between two players is stable, then there cannot be an adjacent network with higher value obtainable by deleting this link. Conversely, for any player not being part of the network, the value that can be added by this player must be smaller than the current value, otherwise the link would be formed.

The concept of pairwise stability requires that at most two players act at the same time, and that the players look only one step ahead. The concept of strong stability extends pairwise stability to coalition of players:

Definition 13. *Strong Stability (Jackson and van den Nouweland 2005). A network* g *is strongly stable with respect to* Y *and* vf *if for* $H \subseteq V$ *and* g' *obtainable from* g *via deviations by* H, *and* $v_i \in H$ *such that* $Y_{v_i}(g', vf) > Y_{v_i}(g, vf)$, *there exists* $j \in S$ *such that* $Y_j(g', vf) < Y_j(g, vf)$.

That is, a network can only be stable if a subset H of players has no incentive to alter it.

For dynamic models of network formation, Jackson and Watts (2002) adapted the concept of stochastic stability (Young 1993). In the dynamic version of the game, at each time step two randomly selected players decide to form or sever a link. The players act myopically and base their decision on whether they are better off with the alteration in t+1. That is, they do not consider the possible consequences that may follow by changing the

utility of other players. After the decision is taken, with some probability $\epsilon > 0$ the alteration is applied, or with $1 - \epsilon$ not. This is a Markov chain with the states being the respective networks that are formed during the process. With $\epsilon \to 0$ the stationary distribution converges to a unique limiting stationary distribution. From this follows the next definition:

Definition 14. *Stochastic stability (Jackson and Watts 2002). A network in the support of the limiting stationary distribution of the dynamic process is stochastically stable.*

Jackson and Watts present methods that allow the identification of stochastically stable networks. The main idea is to identify paths between adjacent networks leading with the smallest possible resistance to a pairwise stable network. Resistance describes whether there exists an improving path from a given network (i.e. with every step all players have to be better off), and if not, how often some deviation from the individual rational choice (described by ϵ) has to be made. More details follow in the next section.

4.3 Models of Network Formation

The JW model is essentially a proof of the existence of stable and efficient networks. Subsequent work based on this model (Watts 2001; Jackson and Watts 2002; Hummon 2000; Doreian 2006) as well as related work (Bala and Goyal 2000; Beal and Querou 2007) provide a dynamic perspective.

In the JW model, players are fully informed, perfectly rational and myopic. Two players can choose at a time to link to each other. The link is only formed if both players agree. The links are undirected since both ends are involved in establishing it. Links can be severed unilaterally; the game is hence partially cooperative. After their decision, the network value

is computed, and the value distributed among the agents according to the equal-split allocation rule. Direct links are costly, and both agents bear the costs of the link. Then, the next two players are selected, who take their decisions based on the current value of the network and the value that would result by their respective actions. As they are myopic they only consider the next state of the network. This process goes on until pairwise equilibrium is reached. Depending on the cost of links, three different equilibria can be sustained: the fully connected network, a sparsely connected network, and the empty network.

The utility function is given by:

$$u_i(g,t) = w_{ii} + \sum_{j \neq i} \delta^{t_{ij}} w_{ij} - \sum_{j:ij \in g} c_{ij} \tag{4.1}$$

t_{ij} is the number of links in the shortest path between individuals i and j. Links between players have a certain value w_{ij}, plus a constant 'intrinsic' value w_{ii} that each player perceives (so that, say, remaining unconnected can have its own utility). $0 < \delta < 1$ is a decay factor by which the value of connections may decrease. $\delta^{t_{ij}}$ captures the fact that the longer the path between the two nodes, the smaller its benefit becomes. If i is not connected to j, δ is set to 0. Direct links are the most valuable, but they come at a cost: c_{ij} denotes the costs of maintaining direct relationships (e.g. time and effort); for all indirect connections, it is set to 0.

For simplicity, Jackson and Wolinsky set w_{ii} to zero and w_{ij} to 1, so that the network depends only on the rate of decay and the cost of direct links. Furthermore, cost and value are dependent on links, not players. Therefore, the indices are left out, and only c and δ is written. They prove the following properties of the network game:

– $c < \delta - \delta^2$: The complete graph is the only unique stable solution.

Players will choose to connect with each other directly provided that the cost of a link is lower than the value gained from it: The value of the highest valued indirect link δ^2 is smaller than the net-value $\delta - c$ gained from a direct link.

- $\delta - \delta^2 < c < \delta$: Many solutions are possible, namely all those benefiting from indirect links. In this case a direct link has positive utility, but as $\delta - c < \delta^2$ it is more beneficial to be indirectly linked. One of the stable solutions is star, as this structure minimises the number of links and the distance between the nodes.

- $\delta < c$: The only feasible solution is the empty network. No player would be willing to create a connection, even if there exists a network that yields positive payoffs.

They also show that for all n, a unique efficient network exists:

- If $c < \delta - \delta^2$ then the complete network is efficient, as the utility of any direct link exceeds the benefit of an indirect link.

- for $\delta - \delta^2 < c < \delta + (n-2)/2 * \delta^2$ the star is efficient. It minimises the number of direct links while connecting all players with a minimal distance.

- for $\delta + (n-2)/2 * \delta^2 < c$ only the empty network is efficient; that is. For any situation where costs exceed the value that can be generated by the star.

Watts (2001) analyses the actual process of forming the network in the connection model. The static model only identified the equilibria and confirms that stable network states exist, but does not reveal whether and how

these can actually be reached. In the dynamic version, two players are selected randomly and given the opportunity to form a link. Players are myopic, and thus anticipate in their decision only the utility of the network that forms in the next step. The process stops if a stable network results.

She finds that two main attractors are possible: The formation of a stable network, or a cycle of adjacent networks (an adjacent network is a network that is obtainable by adding or deleting one link) without any sustainable equilibrium. A network can only be pairwise stable if it can be reached over a path of adjacent networks. Where there is no such path, after some time all feasible networks have been visited, and the process must cycle along those networks. In more detail, the main results are:

- $\delta - c > \delta^2 > 0$: The fully connected network forms. In each period utility strictly increases for any two players not yet directly connected. Since breaking any link an agent reduces his payoff, no links will ever be broken, as in the static model.

- $0 < \delta - c < \delta^2$: Stable non-empty networks can form. The star is efficient and is also a pairwise stable network, although not the unique one. The probability that a star develops decreases as n goes to infinity, because its formation depends on the order in which players meet: Some agent must be the centre agent. If the centre agent C meets another agent A not yet linked to it, then C will agree to establish the connection only if A is not linked to anyone else already connected to C. Otherwise, C would lose the benefit of the indirect link. Thus, the star can only form if all agents meet the centre agent first. The link will be established because with $0 < \delta - c$, any direct link between isolated players will be formed. The larger n, the more likely that unconnected players meet each other before meeting the

same (centre) agent. As a consequence, the likelihood of cycles or the convergence to sub-optimal solutions increases. Especially in the higher cost regions, agents prefer to connect to players who already have a link. As the chance to meet the centre agent first decreases with n, the process is likely to converge to a network with only one path connecting every pair of players (i.e. a 'line network').

– $\delta - c < 0$: No link is formed. Myopic agents cannot form any links, since there is no benefit in establishing the first link, even if connected networks with a utility > 0 do exist.

Jackson and Watts (2002) generalise this approach by modelling it as a stochastic process, Again, two players are selected randomly, but their decision to form or not form a link is only carried out with a certain probability $1 - \epsilon$, whereas with probability ϵ nothing is done. The parameter ϵ may be thought of as errors individuals make in their calculations, or deliberate deviations in order to explore different paths. The smaller ϵ, the more likely the results converge to that of Watts (Watts 2001). However, with larger random perturbations, the myopic nature of the players can be overcome by visiting networks that would not result by rational, myopic decisions. Thereby, a new path of adjacent networks can be reached, possible leading to a pairwise stable network. As already indicated (see definition 14), the dynamics can be formalised as a Markov process on the random variable ϵ. As $\epsilon \to 0$, stable networks that cannot be reached are excluded, and the process selects those solutions that can actually be reached by myopic players. An application to the co-author model (Jackson and Wolinsky 1996) demonstrates that the complete network is selected as the unique stochastically stable network out of several possible solutions. This means other stable solutions might exist, but are not reachable. However, they also

demonstrate that there are examples where all pairwise stable networks are equally stochastically stable.

Hummon (2000) uses the same model specification as Watts (2001), but he simulates the model computationally to obtain his results for n=3, 5 and 10 (see also Doreian (2006) for a detailed, but purely descriptive follow up for n=5 and n=6). The most important observation in this context is that on average in all cost ranges either a star or a ring emerges as the most frequent solution. Which formation occurs depends solely on n and the order in which actors meet. As Watts (2001) derived theoretically, the simulations show that with increasing n the frequency of the star decreases. Only in the lower cost ranges the star still forms.

Bala and Goyal (2000; BG) analyse the formation of communication networks as a non-cooperative game. In the BG model, links can be formed and severed unilaterally. Agents who initiate links have to bear all the costs. They consider two variants of the model, on in which benefits accrue only to the linking agent (1-way-flow model), and one where benefits are shared between players (2-way-flow model).

Using a payoff function without decay, the payoff of a player is given by the benefit received of direct and indirect links minus the cost of direct links in network g:

$$\pi_i(g) = \mu_i(g) - c\mu_i^d(g) \tag{4.2}$$

The marginal benefit of being connected to another agent is normalised to 1. c is the cost, μ is the number of all players player i is connect to, and μ_i^d is the number of direct links the agent maintains.

In any setting where the benefits exceed the costs, it is a best response to link to at least one other player. Bala and Goyal show that in the 1-

way-flow model the Nash equilibrium network is either empty or minimal connected. A minimal connected network is a network in which all nodes are connected and splits apart into more than one component as soon as one link is severed. In the 2-way-flow model, the equilibrium network is either empty or minimally bi-connected, meaning that agents are connected in the form a directed graph, and no redundant links exist.

This equilibrium definition includes a large number of networks as the number of player grows. For example, for three players there are already five Nash networks in the 1-way-flow model. As a refinement, Bala and Goyal define strict Nash equilibrium networks. A strict Nash equilibrium exists if there is no other strategy available for all players i that is a best response given a strategy profile of the other players $_i$. For the 1-way-flow model, Bala and Goyal show that the Strict Nash equilibrium is either the empty network or a wheel (a directed graph in form of a ring). More specifically, if $c < 1$ the ring is the unique equilibrium; if $1 < c < n - 1$ both empty and wheel network are stable; if $c > n - 1$, the empty network is the unique stable network. In the 2-way-flow model, the equilibrium network is either the empty or the centre-sponsored star network (a star where the centre player pays all the links). If $c < 1$, the centre-sponsored star is the unique equilibrium; if $c > 1$, then the empty network is the unique strict Nash equilibrium.

To investigate the question whether these static games actually converge to strict Nash networks in a dynamic setting, the game is specified as a repeated one. The start is a random network, and each player plays his strategy sequentially. All players observe the resulting network as well as the strategies played. Players remain with their last strategy with a probability p, or decide to play new action with probability $1 - p$. In the latter case they decide on a best-response given the actions played by the other players

in the previous round of the game. Bala and Goyal then identify limiting cases of strict Nash equilibria by looking at the changes that are induced when exactly one player adapts his strategy. Simulations are used to test whether the game converges to these limits for different p and to determine the speed of convergence. They find that in the 1-way-flow model, the rate of convergence is rapid, reaching one of the predicted networks in less than 20 rounds. In the 2-way-flow model, convergence takes longer. The smaller p, that is, the closer behaviour to pure best-response, the quicker strict Nash equilibrium is reached. The intuition behind this is that with $p = 0$, the network will oscillate between full and empty networks (assuming an initially empty network) as all agents move. In the first step, linking to any player is the best response. When the network is fully connected, severing all but one link is optimal. With p close to 1, at some stage only one agent will not move, leaving it at the centre of a centre-sponsored star with some positive probability (see also Bernasconi and Galizzi (2005)).

Beal and Querou (2007) model a network game with a notion of bounded rationality. They begin with a one-shot game. In the model, forming a link requires the consent of both players. Players incur costs for *offering* the link; consequently, players only offer links if they know that their opponents do the same. This results in the empty network as unique Nash equilibrium if players are fully rational. In their dynamic version of the game, players have limited memory, but are otherwise perfectly informed about other players' past actions. The game is repeated over a finite number of time steps larger than players' memory. Players maximise their average payoff. Beal and Querou show that with this form of bounded rationality, non-empty networks can exist. Any deviation must be weighted by the players against the potential harm that results from deleting links, as the other players will never link once it has been revealed that the other player does not link until

they forget the deviation until they forget the deviation. As a result, the costs of establishing new links cannot be too high, or the potential value gained from a link must be large enough before any link can emerge.

More recent BG-type models look at the role heterogeneity plays for equilibrium selection. Although heterogeneity is out of scope of this chapter's model, these models are noteworthy because of some experimental results related to them (discussed in section 4.4). McBride (2006) focuses on value heterogeneity and partial information. Value heterogeneity is given if the value of connections is different among players; partial information means that a player observes only the actions of his direct neighbours. In such cases, inefficient outcomes might emerge, whereas under perfect information, the efficient minimal connected networks are also equilibrium networks. Other authors analyse the role of heterogeneous cost for establishing links (e.g. Galeotti et al 2006). They find that in equilibrium state, cost-heterogeneous players form either empty or centre-sponsored star networks; if value varies as well, a strict equilibrium is either the empty network or a minimal connected network with components being connected in the form of centre-sponsored stars.

Models with farsighted players (Watts 2002; Deroian 2003) or coalition formation (Dutta and Mutuswami 1997; Jackson and van den Nouweland 2005; Slikker and van den Nouweland 2000) are related to the network formation game, but use different assumptions about agent behaviour and cooperation among agents. When players are allowed to form coalitions, conditions for equilibrium are stronger and thus reduce the number of possible equilibria since deviations require the consent of all concerned players in the coalition. Using definition 13, Jackson and van den Nouweland (2005) show that strongly stable networks are efficient. When players are farsighted, situations where the costs of links formed exceeds the benefits, but the re-

sulting (non-empty) network has a positive payoff for the connected players, can be overcome. However, although efficient networks could be formed in such cost ranges, this does not happen because each player wants to prevent to become the centre of a star-like structure. Rather, circle networks distributing costs and benefits equally are likely to form.

There have been no applications of RL to strategic network formation games in particular. Using the stag hunt game, only Pemantle and Skyrms (Pemantle and Skyrms 2000; 2004) provide an RL approach of link formation. In the stag hunt game, there are two types of hunters, stag hunters and hare hunters. Both receive a higher utility from hunting with the same types, a lower utility of being in a group with hunters of the other type, and a zero payoff if they stay alone. At each time step, hunters can propose to form a group with two other hunters. Hunters who receive a proposal always accept the offer, so that the group will form if a proposal is made. Starting with equal propensities to form cliques with any type of hunters, the process converges to cliques of the same types if recent experience is weighted higher. On the other hand, if agents remember all their experiences, the process is much more unstable or converges very slowly.

4.4 Experiments with Network Formation

Several network game experiments have been conducted. Most experiments are based on the BG model; only few follow a similar specification as the JW model, which is the focus of the RL model analysed later. However, also for the (partial) cooperative JW model some conclusions can be drawn from the experimental literature.

Vanin (2002) conducts an exploratory experiment of the JW model with four players. The cost setting is $\delta < c < \frac{N-2}{2}\delta^2$, that is, in the medium

range where the star is efficient, but not stable. The value of linking to other players j, w_{ij}, is set to 1000; the cost of a link is 1000; δ was set to 0.8. Pairwise stable is any minimal connected network. Three different groups played the game cooperatively by discussing possible solutions and agreeing on the links they form. A first treatment allows for side-payments to compensate those players bearing larger costs; the second is without side payments. With side-payments two groups coordinate on efficient outcomes, while the third group forms a ring. In the second treatment, there are no side-payments. The first two groups coordinated on the line network. The other group, however, did not consider to agree on an unequal outcome and coordinated on a ring, splitting the cost equally. This result is remarkable insofar as the line is the pareto-optimal outcome: While the ring provides an equal payoff of 240 to all players, the line provides a payoff of 240 to the players with 2 links, but the two extreme players get 952. This agreement was reached tossing a coin. Such an outcome requires that players accept inequality that the players distinguish between the opportunity to gain more before the game starts, and the actual outcome.

Falk and Kosfeld (2003) consider the BG game with 1-way and 2-way flows of benefit and no decay. There are four players in the game. The cost settings cover empty, minimal and star networks as the equilibrium prediction. The game is played for five rounds. Links are formed simultaneously. After a step, players are informed about the network, costs and the connected players. They find that

- In 1-way flow models many outcomes are Nash strict Nash equilibria (between 40 and 60 %). However, for the 2-way flow model, there is no strict Nash equilibrium, and fewer Nash equilibria (between 10 and 30 %).

- If there is more than one unique stable network, subjects solve the coordination problem by opting for the efficient network.
- Higher costs support the selection of both Nash and strict Nash solutions in the 1-way flow model, but have a negative impact on the selection in the 2-way flow model.

Falk and Kosfeld (2003) provide two possible explanations for the unequal results in the 1-way and 2-way flow models. The first possibility is the asymmetry in payoffs: In the 1-way flow model, the ring is the stable network, as each player has to create a link to participate in the value of the network. Costs and benefits are distributed equally. In the 2-way flow model, the stable solution is the centre-sponsored star, but no rational player wants to be in this position. Their data support this hypothesis, as they find that when such solutions are reached, they are unstable, i.e. the disadvantaged players sever their links. The other possible explanation offered are social preferences. This hypothesis is supported by their finding that the frequency of Nash outcomes decreases the more unequal the payoffs are - this becomes especially apparent in the low frequency of the centre-sponsored star. Using a regression model, they find that individuals are more likely to revise their strategy if outcomes were unequal.

Using a similar setup as Falk and Kosfeld (2003), Bernasconi and Galizzi (2005) find very different results. They consider four treatments with low and high costs and one- and bi-directional flow of benefits. The main difference to the former experiment is a more neutral labelling. Bernasconi and Galizzi (2005) claim that the use of ordered labels A,B,C,D in Falk and Kosfeld's experiment serves as a coordinating device, as they find in their own experiments that the ring from A to D can be observed significantly more often than when random labels are used. They therefore choose instead

more neutral labels like '&' or '%'. They find that in the one-directional treatments almost no Nash networks emerge (between 1% and 3%). In the bi-directional experiments sometimes Nash networks form, but also with comparatively low frequency (between 13% and 17%).

Callander and Plott (2005) consider a BG model with 1-way flow of benefits with no decay. They consider different treatments with homogenous and heterogeneous cost settings. Cost settings are such that the wheel is strict Nash. For the homogenous case, they find that

- The empty network never occurs.
- If networks converge, it is usually a Nash equilibrium, however, not strict.
- Not all Nash equilibria are stable, often an equilibrium state collapsed again.

Looking at how players take decisions and the dynamics of behaviour, they find that

- Players do typically not play myopic best-responses as in the BG model. They often use simple strategies considering the future outcomes of the game. Agents make more sophisticated decisions anticipating future outcomes.
- Agents using such simple strategic behaviour follow their strategy more consistently.
- Convergence depends on the behaviour of all agents. The more agents switch to simple strategic behaviour, the more likely the network converges.

– The more agents remain committed to their behaviour, the more likely other agents will adopt this behaviour as well.

Conte et al (2009) investigate a link formation game where links are formed only if both players agree. In each round of the game players bid for links simultaneously. The main interest is not whether networks converge, but which individual strategies are responsible for the result. There are six players, no decay, and cost settings are such that the equilibrium prediction is a minimal connected network. Subjects have full information about the network. In total, there were 54 participants. Nine experimental sessions were run with six players per session. A session lasts at least 15 rounds, after which a random generator determined to stop the session. In the experiments, minimally connected networks emerge; however, stability is low. Conte et al (2009) attribute this to the fact that many equilibria are possible, so that it is difficult to coordinate on a certain outcome. They also observe that when a minimal connected network is established, some players are tempted to experiment with alternative strategies. As a result, a network might come out of equilibrium again. From the individual perspective, they find that 40 % of strategies are best-response strategies. The remaining 60% strategies are not very far from best-response behaviour. Distance is determined by calculating an index based the difference between profits of actual and best response behaviour. Common alternative strategies are reciprocator and opportunistic behaviour. The first behaviour maximises direct connections by always offering links to those players who offered links in the previous round. The second behaviour tries to maximise indirect links by removing direct links whenever possible. Best response behaviour is strongly group driven, i.e. the more players adopt this strategy the more likely that the remaining agents follow. There is an overlap between best response and the other strategies. Conte et al (2009) estimate

econometrically that 42% of players belong to the opportunistic, 31% to the best response type and 27% to the reciprocator type. The high portion of the opportunistic type thus points, similar as the previous studies, to more complex than myopic best response behaviour.

Goeree et al (2009) test whether heterogeneous players manage better to agree on efficient networks. They consider three treatments: A baseline treatment with homogenous agents, a treatment with a low-cost agent (experiencing lower costs for maintaining a link), and a treatment with a high-value agent (experiencing and providing higher utility per direct or indirect link). They find that with homogeneous agents, formation of equilibrium networks fails. Introducing cost heterogeneity supports the emergence of equilibrium networks in the form of minimal connected or star-networks. When agents receive different value from linking the chance to observe equilibrium networks is highest.

Summarising the main results of the experimental literature, the following conclusions can be drawn:

- The frequency of equilibrium networks differs strongly between the experiments. Some authors find no Nash networks at all. Maximum rates observed go up to 40%.
- Even where Nash networks are found as good predictors, it becomes apparent that the actual individual decisions deviate from the myopic best response (Callander and Plott 2005). Basic strategies like opportunistic linking, reciprocating behaviour or simple strategic-decision making are more common.
- The more agents commit to a certain behaviour, the more likely convergence.

– Some authors further mention an equality norm, i.e. a preference of the players for equal distributions of cost (e.g. Vanin 2002).

4.5 A Reinforcement Learning Model of Network Formation

One conclusion of the literature review is that actual human behaviour in the experiments differs often from the equilibrium condition. This section describes the RL based model of network formation and asks how the outcome differs from the theoretic predictions.

A dynamic version of the connections model is considered, similar to Watts (2001) and Jackson and Watts (2002), but adapted to a setting with RL agents. As a benchmark, the original analysis of Jackson and Wolinsky (1996) for the static, and Watts (2001) for the dynamic model can be used.

The game proceeds as follows:

– Two agents are picked randomly.

– Both agents decide whether to offer a link or not.

– If both agents offer a link, the connection is added, otherwise not.

– The new network is computed.

– The two agents who acted receive their rewards, calculated with equation 4.1.

If a link was formed, it exists as long as the two agents do not meet again. When they meet another time, the link is maintained if both agents offer a link again, otherwise it is severed.

Learning In the reviewed network games literature bounded rationality was described as an injection of 'irrationality', for example, as error term ϵ as in Jackson and Watts (2002) or Bala and Goyal (2000), or a limited memory as in Beal and Querou (2007).

In the model presented here, RL can be seen as a form of limited rationality. Agents start with no information at all and learn by trial and error about the game and the application of the appropriate actions. Players know only the name of the other players and may choose from the action set A $=\{a_0 \dots a_i \dots a_n\}$ given by {offer link, not-offer link}.

Using the concepts of BRA introduced in chapter 2, the internal choice model for agent i is given by $r^k_{u,v} : C^{k,k\neq i}_{u,v} \rightarrow \{\text{offer}, \text{not-offer}\}$. There are $k-1$ mappings and the initial conditions contain only one attribute with one value (player-name=k), so no further expansion is possible. BRA thus reduces to disjoint sets of simple RL rules. For each r^k agent i updates the action strengths, that is, $\forall r^k$, using

$$q(a_j(t,k)) = q(a_j(t-1,k)) + \gamma(u_i(g,t) - q(a_j(t-1,k)))$$

Using the exponential selection rule in equation 2.10, agent i chooses at the next encounter with agent j his action.

Parameter settings The model has four parameters of interest, α and γ, cost c and value δ. As in the original JW model, w_{ij} is set to 1, and w_{ii} to 0. Agents are homogenous; cost and value are the same for all players.

In the simulations, the parameters c and α are varied. c can be seen as the structural parameter influencing the opportunities for the players; α determines the rate of exploration. The greater α, the more likely exploration in the action selection process and the selection propensities for both actions become more similar; the smaller α, the faster the agents stick to

a reasonably good solution. The central question for the adaptive network model is whether it is possible to generate stable and efficient solutions, and how the properties of the learning rule have to be for this. The influence of randomness on the outcome has led to the choice of stochastic stability as the benchmark stability definition for the RL model.

The discount parameter γ is only of minor importance for the analysis. γ sets the rate at which the reward is updated. The smaller this weight, the faster the experienced reward approximates the true reward. Experiments with various γ values were used to select the best model for a more detailed analysis of α. A short overview of different γ settings is given in section 4.6.4.

The value of δ, $0 < \delta < 1$ is fixed at a value larger > 0. Since there are no requirements or other substantial reasons for a particular value except that decay exists, it has been set to 0.5. For each cost range, the values for c are drawn randomly in order to obtain some samples within each cost range. α is incremented by 0.01 from 0.01 to 1.

Table 4.1 shows the parameters in summary.

cost range	α	δ	γ
$c < 0.25$ (low cost range)	0.01 ... 1	0.5	0.1, 0.25, 0.75, 1
$0.25 < c \leq 0.5$ (medium cost range)	0.01 ... 1	0.5	0.1, 0.25, 0.75, 1
$c > 0.5$ (high cost range)	0.01 ... 1	0.5	0.1, 0.25, 0.75, 1

Table 4.1: RL network model parameter settings

Measurements for the simulations Networks and network formation can be described in a variety of ways. In section 4.2.1 the measures D (density) and L (average path length) were already introduced. Three additional measures are defined here:

A stability measure is computed to assess how robust the solutions are. It might occur that a simulation result comes very close to the theoretic equilibrium in settings where agents explore enough and discover better solutions. Since exploration comes at the cost of more random decisions in the process, the whole system can become unstable.

Definition 15. *Stability.* $S_t = 1 - \frac{1}{2}\frac{n_{(g,t-1)} - n_{(g,t-1)}}{(n(n-1))}$

Stability is simply the difference in the number of links between two time steps, divided by the number of maximum possible links to standardise the measure. For a single simulation step, the value can be either 0 or 1. Over a sample of simulations, S_t can be interpreted as the probability that a link changes at t. It thus varies between 0 and 1 and the closer it is to 0 the more stable the network is.

To compare the results with the game-theoretic prediction, a fitness measure is defined as follows:

Definition 16. *Fitness / Efficiency. Let the vector* g_{stable} *be the stochastic stable network (efficient network), and* g_{actual} *a simulated network. Let* $steps_{max}$ *be the maximum number of modifications starting from any network to* g_{stable}*, and* $steps_{actual}$ *the number of modifications to reach* g_{stable} *from* g_{actual}*. Define the fitness at time t as:* $fit_t = \frac{1}{2}(\frac{steps_{actual,t}}{steps_{max}} + \frac{steps_{actual,t}}{steps_{max}} S_t)$

The resulting measure varies between 0 and 1 and tends towards 1 the closer the network structure to the stochastic stable network, and the more stable the simulation result (multiplying the distance with S_t and adding it in the enumerator has the effect that stable states are weighted higher as $S_t = 0$ if a linked changed, 1 otherwise).

To determine the stochastic stable network, the procedure in Jackson and Watts (2002) has been implemented as a computer program. The program computes the set of all possible networks, and finds out the pairwise stable network with the minimal resistance from all other networks in the set.

4.6 Simulations

4.6.1 Overview

Simulations were run for at least 10.000 time steps per α and γ combination for each cost range in samples of up to 4000 steps with several repetitions per simulation, giving a reasonably large sample. Figure 4.1 shows how fitness values vary depending on α and γ.

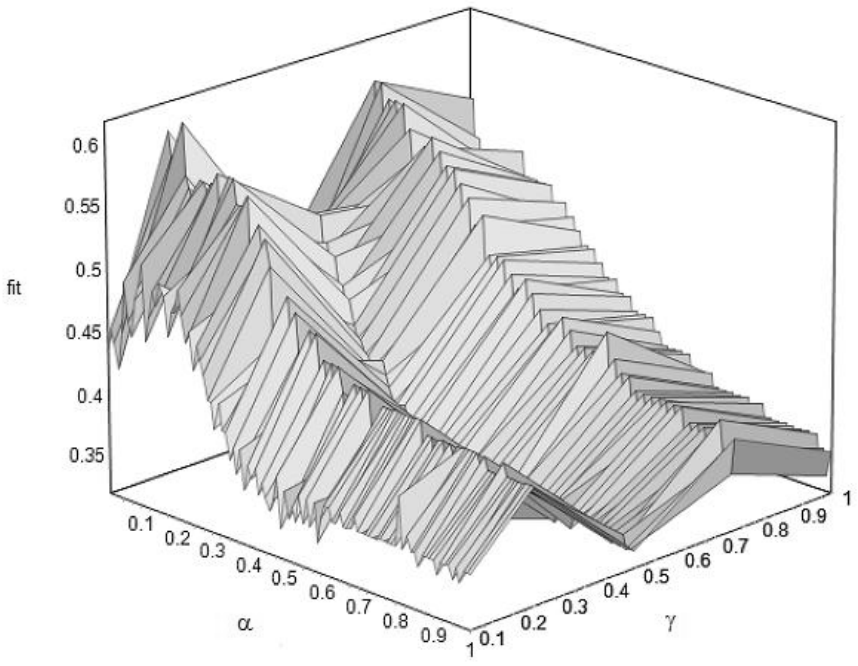


Figure 4.1: Fit of simulations for simulated α and γ values.

Across all γ, the fit of the simulation develops analogously - starting with a high fit of up to 0.6 for small α, then decreasing to values of about 0.35. Furthermore, results fit better for $\gamma = 0.25$ and $\gamma = 0.75$; that is, for values close to very long memory and no memory at all. The reason for this behaviour lies in the different role of adjustment speeds to other agents' behaviour depending on the cost range. This is discussed in section 4.6.4.

Analysis revealed that the γ and α combinations maximising fitness in each cost range are $\alpha = 0.1$ and $\gamma = 0.75$ for the low, $\alpha = 0.01$ and $\gamma = 0.25$ for the medium, and $\alpha = 0.07$ and $\gamma = 1.0$ for the high cost range (in the high cost range several combinations achieve a fit of 1. Out of the top 20 results, the simulation belonging to the most frequent γ value and the highest α was chosen). Some more samples for these specific values were simulated to look closer at the behaviour for various cost values. Using network density as an indicator, figure 4.2 illustrates connectivity as a function of cost.

In the high cost range ($c > 0.5$) the empty network emerges as solution. In the low cost range ($c < 0.25$), connectivity is high (almost fully connected structures). In the medium cost range ($0.25 < c \leq 0.5$) networks become sparser (density between 0.4 and 0.5). For the 'border' regions between low and medium, as well as medium and high cost range connectivity changes gradually; for example, for cost=0.54, density was 0.28. In the RL process, no threshold function between cost ranges emerges as stated in the benchmark model.

The following sections analyse the behaviour of the simulation in more detail. It is analysed how the shape of networks changes when α changes. The question is how the exploration and exploitation affect the connectivity and stability of small networks. γ is held constant at the value maximising

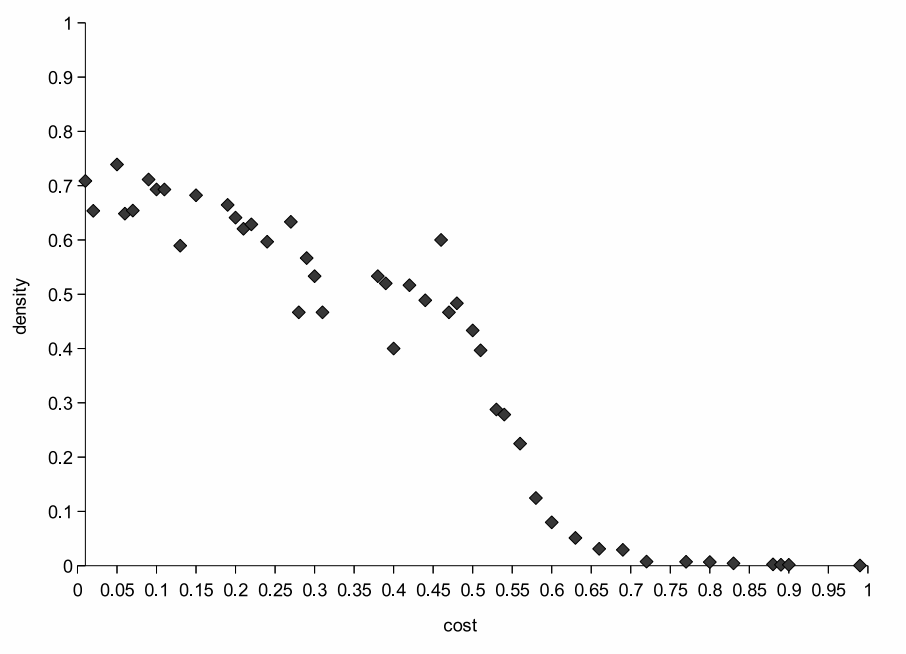


Figure 4.2: Network density over all cost samples.

the fitness in each cost range.

4.6.2 Network Properties for Different α

Figures 4.3, 4.4 and 4.5 show how density, stability and fit develop in the low, medium and high cost ranges.

Low cost range ($c < \delta - \delta^2$) The optimal solution is the fully connected network ($D = 1$). Figure 4.3 shows that for small $\alpha(\lessapprox 0.07)$ the network is strongly connected ($D \approx 0.7$) without reaching the complete network, and stability tends towards 1. As could be expected, for small α, agents tend to stick to first-best solutions, which are those providing the largest increase in marginal utility. With α increasing towards ≈ 0.11, the network is developing towards the fully connected network ($D \approx 0.8$). However, this comes at the cost of stability, i.e. some agents keep switching. Finally,

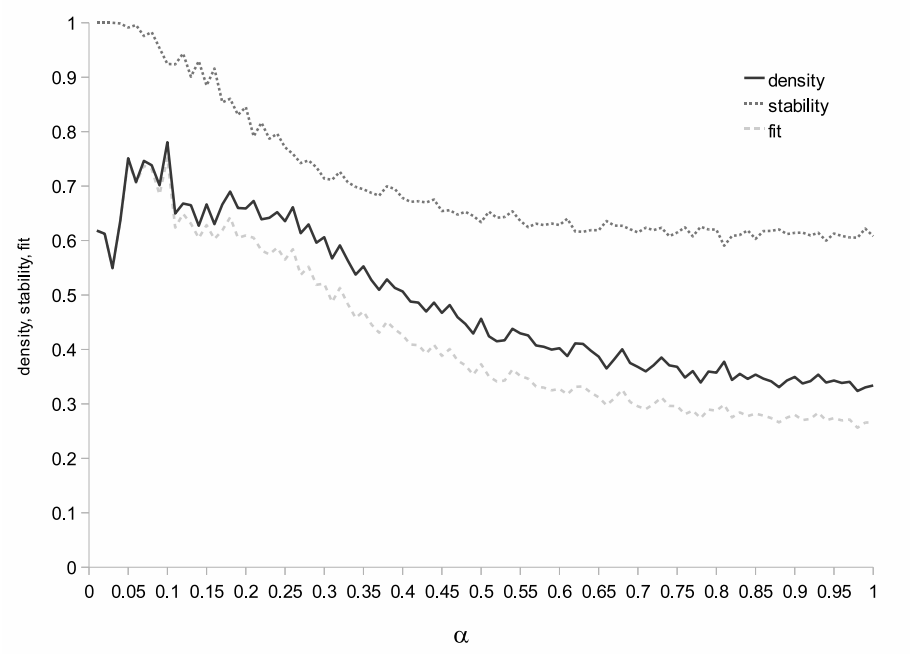


Figure 4.3: Network density, stability and fit for the low cost range.

for $\alpha > 0.6$, connectivity and variability approach a limit in an asymptotic manner with density about 0.35. The random limit is given by the probability that offered links are accepted. Assuming total randomness, the chance of offering a link is 0.5, the chance that the other player offers a link at the same time is equally 0.5. Thus, the probability that a link can actually be formed by pure chance is 0.25. This indicates that RL performs better than randomness, even if the distance between the action propensities becomes smaller.

Medium cost range ($\delta^2 < c \leq \delta$) According to the static as well as the dynamic model, minimal connected networks should form (i.e. $D \approx 0.5$). Computations showed that the star is the efficient as well as stochastic stable pairwise network. In the simulations, agents end up very close to a minimal connected network ($D \approx 0.5$) for $\alpha < 0.06$. These networks

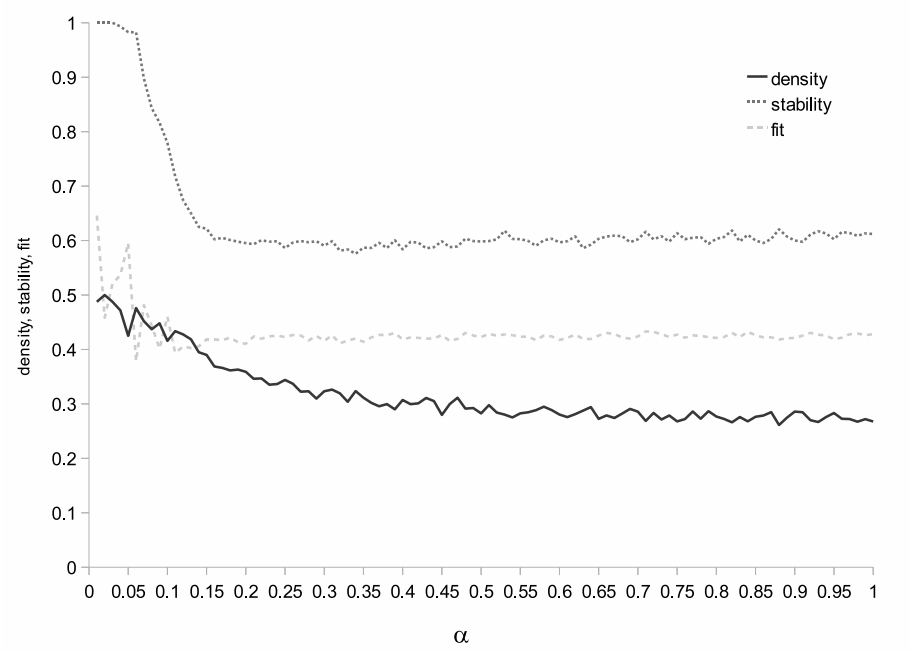


Figure 4.4: Network density, stability and fit for the medium cost range.

are very stable. For $0.07 < \alpha < 0.15$ there is a decrease in density to ≈ 0.4, with a sharp drop in stability and corresponding decreases in fit. For $0.15 \leq \alpha < 0.3$ density decreases further. For $\alpha \gtrapprox 0.3$ the connectivity of the network settles asymptotically near to the random limit; similar to the low cost range the RL process performs also here (slightly) better than random. The density of ≈ 0.4 in the range $0.07 < \alpha < 0.15$ indicates that networks are not over-connected, but may be rather efficient. The sharp decrease in stability points, however, to coordination failure (random switching) rather than reinforcements. In principle, optimal network structures can develop simply because they are closer to a random outcome.

High cost range ($c > \delta$) Here, the empty network is expected. Although for some agents positive utility could be generated by indirect links, there is always at least one agent for whom the costs exceeds the value it receives

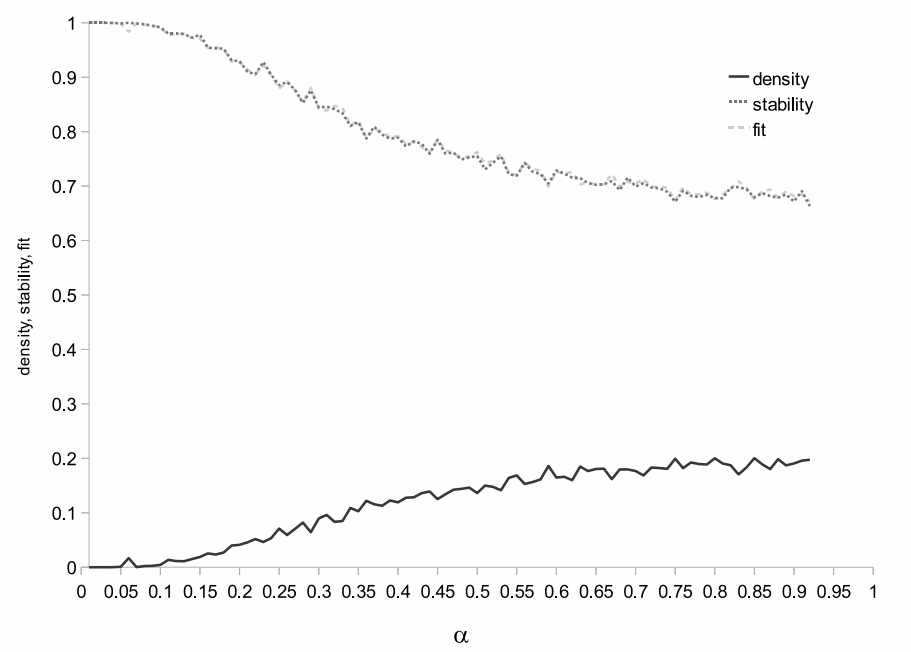


Figure 4.5: Network density, stability and fit for the high cost range.

and thus motivates the deletion of direct links. Figure 4.5 shows that the simulation converges to the equilibrium prediction if agents explore little ($\alpha <\approx 0.15$). For $\alpha > 0.25$, at least two agents are linked ($D = 0.1$). The random limit is approached for α values > 0.3. the model approaches quite fast a situation where at least two agents are linked. At $\alpha \approx 0.6$ the simulation converges to the random limit ($D \approx 0.2$). Here again RL clearly performs better than a random process.

4.6.3 Network Structure and Dynamics

Tables 4.2, 4.3 and 4.4 show the results for the three cost ranges for the measures density (D), stability (S), and match (fit). A cluster analysis for α has been performed on the variables D and S to group the results. These variables have been chosen because they describe the dimensions structure as well as time. For the resulting clusters, the emerging networks

are characterised by network structure, and the summary measures density D, average path length L, fitness fit, S and efficiency E. The choice of three cluster centres reflects roughly the main dynamics observed in figures 4.3, 4.4 and 4.5: A good fitness in the lower α regions, then decrease in fitness (which may mean a decrease or increase in density), finally approximation of the random limit. The tables illustrate the most network architectures which result during this process. For readability, only the upper quartile is represented. The share of each network is based on the frequency in the quartile (not the overall occurrences).

Low cost range ($c < \delta - \delta^2$) Table 4.2 shows the following: In cluster 1 (the cluster with the best fit), the most common visited networks are 2,3,3,4,4; 2,2,2,3,3 and 2,2,3,3,4 with a relatively high connectivity ($D = 0.6$ - 4 missing links to the complete network; and 0.8 - 2 missing links to the complete network). The path lengths of 1.5-1.75 indicate that most networks are connected in a way that each player can be reached directly or with one intermediary at maximum. In the second α range, the most frequent networks 2,2,2,3,3; 2,2,3,3,4 and 1,2,2,2,3 are still connected more densely than sparse networks, but are also quite unstable ($S \approx 0.7$ as compared to ≈ 0.9 in the first cluster). Finally, cluster 3 illustrates that with $\alpha \to 1$, network density approaches its random limit 0.25, with frequent unconnected networks (i.e, $L = 0$).

Medium cost range ($\delta^2 < c \leq \delta$) In the medium cost range, relatively stable networks close to minimal connected networks form. The network 1,2,2,2,3 is the most common one, with an average path length of 2.04, meaning that now often at least one intermediary connects two different players. This is close to a ring (only one player has more links), which is the structure minimising the costs, at the same time distributing them

Cluster		*Network*	*D*	*L*	*S*	*Fit*	*Share*
α	$0.01 - 0.24$	1,2,3,3,3	0.6	1.88	0.92	0.58	0.07
avg(D)	0.68	1,2,2,2,3	0.5	2.05	0.92	0.48	0.09
avg(L)	1.67	2,3,3,3,3	0.7	1.63	0.93	0.67	0.1
avg(S)	0.93	3,3,4,4,4	0.9	1.38	0.93	0.87	0.11
avg(Fit)	0.66	1,2,2,3,4	0.6	1.75	0.93	0.58	0.12
avg(E)	0.66	2,3,3,4,4	0.8	1.5	0.92	0.77	0.15
		2,2,2,3,3	0.6	1.75	0.94	0.58	0.17
		2,2,3,3,4	0.7	1.63	0.93	0.68	0.18
α	$0.25 - 0.46$	1,1,2,2,2	0.4	2.22	0.66	0.33	0.08
avg(D)	0.59	2,3,3,4,4	0.8	1.5	0.75	0.7	0.1
avg(L)	1.84	1,1,2,3,3	0.5	2	0.7	0.42	0.1
avg(S)	0.71	1,2,3,3,3	0.6	1.88	0.71	0.51	0.11
avg(Fit)	0.51	1,2,2,3,4	0.6	1.75	0.72	0.52	0.14
avg(E)	0.51	2,2,2,3,3	0.6	1.75	0.73	0.52	0.15
		2,2,3,3,4	0.7	1.63	0.74	0.61	0.15
		1,2,2,2,3	0.5	2.06	0.69	0.42	0.18
α	$0.47 - 1.0$	1,2,2,2,3	0.5	2.06	0.62	0.4	0.07
avg(D)	0.26	1,1,1,2,3	0.4	2.25	0.62	0.32	0.08
avg(L)	0.51	1,1,1,1,2	0.3	0	0.61	0.24	0.08
avg(S)	0.63	1,1,2,2,2	0.4	2.18	0.62	0.32	0.09
avg(Fit)	0.21	0,1,1,1,1	0.2	0	0.63	0.16	0.1
avg(E)	0.21	0,1,1,2,2	0	0	0.62	0.24	0.17
		0,0,0,1,1	0.1	0	0.65	0.08	0.19
		0,0,1,1,2	0.2	0	0.64	0.16	0.21

Table 4.2: Low cost range network structures

evenly so that no incentives for deviation exist. This is similar to the results of Watts (2001), and - for the non-cooperative game - of Bala and Goyal (2000). For example, in cluster 1, the ring has a share of 0.09. More efficient structures (1,1,2,2,2; 1,1,1,2,3) are more common. Unconnected networks occur already in cluster 1, and become more frequent in clusters 2 and 3; thus indicating that any equilibrium-like state in this cost range is more unstable and difficult to sustain. Whereas D indicates a relatively close match with pairwise stable networks (these are: 1,1,1,1,4; 1,2,2,3,4;

Cluster		*Network*	*D*	*L*	*S*	*Fit*	*Share*
α	$0.01 - 0.2$	1,2,2,3,4	0.6	1.75	0.85	0.62	0.07
avg(D)	0.46	1,1,2,3,3	0.5	2	0.78	0.44	0.07
avg(L)	1.78	2,2,2,2,2	0.5	1.88	0.94	0.16	0.09
avg(S)	0.83	2,2,2,3,3	0.6	1.75	0.85	0.31	0.14
avg(Fit)	0.44	0,1,1,2,2	0.3	0	0.72	0.43	0.14
avg(E)	0.44	1,1,2,2,2	0.4	2.37	0.81	0.3	0.15
		1,1,1,2,3	0.4	2.25	0.88	0.63	0.17
		1,2,2,2,3	0.5	2.04	0.85	0.46	0.22
α	$0.21 - 0.4$	1,2,2,2,3	0.5	2.06	0.54	0.39	0.09
avg(D)	0.24	1,1,1,2,3	0.4	2.25	0.57	0.52	0.09
avg(L)	0.43	1,1,1,1,2	0.3	0	0.59	0.4	0.09
avg(S)	0.5	0,1,1,1,1	0.2	0	0.62	0.27	0.1
avg(Fit)	0.38	1,1,2,2,2	0.4	2.16	0.56	0.26	0.11
avg(E)	0.38	0,0,0,1,1	0.1	0	0.66	0.41	0.14
		0,1,1,2,2	0.3	0	0.6	0.4	0.19
		0,0,1,1,2	0.2	0	0.63	0.54	0.19
α	$0.41 - 1.0$	0,1,2,2,3	0.4	0	0.56	0.52	0.07
avg(D)	0.25	1,1,1,2,3	0.4	2.25	0.56	0.52	0.07
avg(L)	0.32	1,1,2,2,2	0.4	2.15	0.55	0.26	0.08
avg(S)	0.62	1,1,1,1,2	0.3	0	0.59	0.34	0.08
avg(Fit)	0.35	0,1,1,1,1	0.2	0	0.63	0.27	0.11
avg(E)	0.35	0,1,1,2,2	0.3	0	0.6	0.4	0.17
		0,0,0,1,1	0.1	0	0.68	0.42	0.19
		0,0,1,1,2	0.2	0	0.64	0.55	0.22

Table 4.3: Medium cost range network structures

1,3,3,3,4; 2,3,3,3,3; 2,2,2,3,3; 1,1,2,2,4; 2,2,2,2,2 for cost closer to the low cost limit, plus the more sparse structures 1,2,3,3,3; 1,1,2,3,3; 1,2,2,2,3; 1,1,1,2,3 for costs closer to the high cost range), the distance to the unique stochastic stable network 1,1,1,1,4 is larger as compared to the low cost range. That is, while rational myopic players according to the stochastic process of Jackson and Watts (2002) are most likely to end up with a star network, the RL process diverges strongly from this result.

Cluster		*Network*	*D*	*L*	*S*	*Fit*	*Share*
α	$0.01 - 0.27$	1,1,1,2,3	0.4	2.25	0.41	0.42	0.01
avg(D)	0.05	0,1,1,1,1	0.2	0	0.76	0.7	0.02
avg(L)	0.09	0,1,1,2,2	0.3	0	0.7	0.59	0.03
avg(S)	0.91	1,1,2,2,2	0.4	2.39	0.65	0.5	0.04
avg(Fit)	0.9	0,0,1,1,2	0.2	0	0.77	0.71	0.04
avg(E)	0.9	0,0,0,1,1	0.1	0	0.88	0.85	0.14
		0,0,0,0,0	0	0	0.99	0.99	0.72
α	$0.28 - 0.51$	0,1,1,1,3	0.3	0	0.65	0.58	0.02
avg(D)	0.08	1,1,1,1,2	0.3	0	0.64	0.57	0.02
avg(L)	0	0,1,1,2,2	0.3	0	0.64	0.57	0.04
avg(S)	0.75	0,1,1,1,1	0.2	0	0.72	0.69	0.07
avg(Fit)	0.76	0,0,1,1,2	0.2	0	0.72	0.69	0.16
avg(E)	0.76	0,0,0,0,0	0	0	0.89	0.95	0.31
		0,0,0,1,1	0.1	0	0.81	0.81	0.38
α	$0.51 - 1.0$	1,1,2,2,2	0.4	2.13	0.55	0.47	0.03
avg(D)	0.16	0,1,1,1,3	0.3	0	0.62	0.57	0.04
avg(L)	0.05	1,1,1,1,2	0.3	0	0.61	0.56	0.05
avg(S)	0.71	0,1,1,2,2	0.3	0	0.62	0.67	0.1
avg(Fit)	0.72	0,1,1,1,1	0.2	0	0.68	0.91	0.11
avg(E)	0.72	0,0,0,0,0	0	0	0.82	0.57	0.15
		0,0,1,1,2	0.2	0	0.68	0.67	0.22
		0,0,0,1,1	0.1	0	0.75	0.79	0.32

Table 4.4: High cost range network structures

High cost range ($c > \delta$) In the first, cluster the most frequent network is the empty network with a share of 0.73. In the most frequent non-empty network only two players are connected. In the other clusters, non-empty networks are more frequent. In the second cluster, two players link most of the time; in the third cluster it might happen that even more than two players connect.

Cost range	γ	D	L	S	*fit*
$(c < \delta - \delta^2)$	0.1	0.68	1.66	0.95	0.66
	0.25	0.61	1.78	0.96	0.6
	0.5	0.64	1.67	0.83	0.59
	0.75	0.79	1.5	0.92	0.75
	1	0.72	1.6	0.89	0.63
$(\delta^2 < c <= \delta)$	0.1	0.51	2.05	1	0.44
	0.25	0.47	2.08	1	0.61
	0.5	0.51	1.8	1	0.48
	0.75	0.51	1.88	0.99	0.46
	1	0.46	1.93	0.99	0.51
$(c > \delta)$	0.1	0	0	0	0.99
	0.25	0	0	1	0.99
	0.5	0	0	1	1
	0.75	0	0	0.99	0.99
	1	0	0	1	1

Table 4.5: Simulation results for various γ

4.6.4 Memory Effects

To round up the analysis, summary measures are reported for simulation runs with different γ values while holding α constant. For each cost range, the optimal α values were chosen: 0.1 in the low, 0.01 in the medium, and 0.02 in the high cost range.

Table 4.5 shows that in the low cost range fitness and connectivity are best for the higher γ values. Moreover, a γ value of 1 increases connectivity as compared to smaller values. It also affects the stability of the network, as the probability of deviations is the highest. $\gamma = 0.75$ seems to compromise well between exploration, on the one hand, and stability on the other.

In the medium cost range, $\gamma = 0.25$ is optimal. Higher γ values, but also $\gamma = 0.1$, are also here responsible for higher density - which is inefficient in this scenario. Furthermore, $\gamma = 0.1$ and $\gamma = 0.25$ both maximise the path

length, which means they are support networks that connect the players in the sparsest way. As noted above, in the medium cost range utility might strongly decrease after a certain threshold is reached. If agents react very quickly, this could lead to a collapse of the network. More tolerance on the other side might support experimentation on the fringes.

In the high cost range there seems to be no influence of γ (at least not for the chosen α values) - all solutions are typically empty and very stable networks.

4.6.5 Summary of the Simulation Results

Low cost range ($c < \delta - \delta^2$) The likely reason that the network does not approach full connectivity is the decreasing rate of utility the more connected the network becomes. At the beginning of the process, the first links provide the highest marginal utility and reinforce the highest action strengths. After agents are connected directly or indirectly to all other agents (i.e. via flower networks contracting the distance with very few additional links), the marginal utility of exchanging an indirect for a direct link is small. Consequently, the selection probabilities for forming and not forming the link become for certain players more equal the later they interact in the formation process. As a result, the decisions would switch between offering and not offering a link for some of the players, irrespective of α. The situation can, nevertheless, stabilise early in the simulation if a player first experiences either linking or not linking as negative (or 0), but benefits from an indirect link added by another pairing of players. If the distance becomes small enough, the particular action played at that time becomes reinforced, and with α sufficiently small, will be repeated. If α is large, this could result in a cycle where most of all players are at some stage the 'marginal' agent that is not worth linking to. This can be inferred from

the trends in density and stability: For the smallest α values stability is highest, but not density. As α increases, stability decreases stronger than density increases. Moreover, the distribution of visited network structures does not change very much, which means that similar network structures exist during the whole run, but with more frequently changing links. The optimal γ value of 0.75 indicates, furthermore, that agents have a short memory and so react quickly to changes in the network structure.

Medium cost range ($\delta^2 < c \leq \delta$) Up to the level where the utility of not being linked is smaller than being linked, the learning process follows the same marginal utility dynamics as in the low cost range. Once utility becomes negative, the average rewards decrease strongly and prevents further linking. Thus, the cost settings act as a natural cut-off to the reward perceived by the agents. In the low cost range, there is no such bound, but the additional utility becomes very small, leading to random switching. The closer cost to $\delta - \delta^2$, the more similar behaviour in the medium cost range becomes to behaviour in the lower cost range - density increases. Note furthermore, that the optimal γ is with 0.25 very low as compared to the other cost ranges, which means that agents are more tolerant of deviations. A plausible reason for this is that agents must not be 'too' myopic, since for stable networks in this range agents have usually to link to two other agents. The utility of just one link is small and thus the motivation to alter that link is large. Allowing some tolerance for such behaviour ensures that the network does not collapse quickly as a consequence of a single agent severing a link.

Moreover, fit in the medium ranges is worst as the star is the stochastic stable network, but the emerging structures are ring-like. This coincides with Watts (2001)'s prediction that the formation of stars becomes unlikely

the larger n, but conflicts with the stochastically stable star that was computed using the approach in Jackson and Watts (2002).

High cost range ($c > \delta$) There is nothing surprising in the high cost range; it largely reflects the equilibrium prediction. Here, the learning task for the agents is extremely simple because there are very few non-empty networks in which an agent can experience a positive reward. Thus, the only deviation in this case from the prediction is induced solely by the increased randomness in the action probabilities with increasing α. Furthermore, the optimal γ value of 1 shows that the best performing agents react very quickly with no memory at all to alterations in the network structure. This is plausible, since independent of the history, any addition of a link has always a negative impact for at least one agent - which was also stated in the dynamic benchmark model.

Thus, the networks evolving from the learning model differ with the exception of the high cost range quite considerably from the equilibrium prediction. A closer look at the data showed that for the optimal α and γ values (0.1/0.75, 0.01/0.25 and 0.07/1), the pairwise and stochastic stable outcome was met with a rate of 0.01 in the low range (4,4,4,4,4 the only pairwise/strongly and stochastic stable network) as compared to a rate of 0.13 of the most frequent network 2,2,3,3,4; in the medium cost range 51% of all visited networks were pairwise stable, but only 19% stochastic stable; only in the high cost range 71% of all networks were the predicted empty network. Looking at the structure of the networks that evolved, it is more accurate to speak of two characteristic cost ranges, one with $c < \delta$ and one with $c > \delta$. In the ranges where positive utility is achievable, agents form sparsely connected networks, adding some shortcuts contracting the distance between them (flower networks). The smaller cost, the closer the

resulting networks are to the complete network. The higher the cost the more sparse the resulting network will be - independent of whether the cost is in the low or medium range. The shorter the distance between agents in the network, the more undecided agents become whether to connect to some other player directly or not. If $\delta^2 - \delta < c < \delta$, the RL process matches pairwise stable networks more often because utility is increasing with the first additional links, but later decreasing (i.e. marginal utility is in the very low cost range convex, whereas in the second case, it is decreasing after reaching its maximum). Another factor is simply chance - sparse networks are simply closer to the random limit of 0.25.

4.7 Applying BRA

In the base model, knowledge is pre-wired - agents maintain a state-action mapping per player and form expectations about the behaviour of each player. The implicit assumption was that learning is simplified by saving the necessary specialisation and generalisation procedure. It thus helped to reduce complexity, and concentrate on the effects of pure RL in a network game context. Applying BRA as described in chapter 2 and allowing to evolve this internal model dynamically can be seen as a further test of robustness - is it possible to perceive player-specific behaviour (similar to the discrimination game), and if not, does this impact the result at all?

In the BRA network model, agents develop the state-action mappings themselves. The initial rule has the form $r_{0,1} : C_{0,1} \rightarrow A$ where the condition can be described with: (player-name=2 *or* player-name=3 *or* player-name=4 *or* player-name=5) for the first player, for the second player (player-name=2 *or* player-name=3 *or* player-name=4 *or* player-name=5) and so on. During the process of the simulation, agents expand this initial

rule into finer grained mappings, for example $C_{1,1} \rightarrow A$ with $C_{1,1}$ (player-name=2 *or* player-name=3) and $C_{1,2} \rightarrow A$ with $C_{1,2}$ (player-name=4 *or* player-name=5). The idea is that with this mechanism the base model can be learnt if the distinction per player label is useful.

Three simulations, one for each cost range, were run for 5000 time steps each. Parameters where set as follows: $\chi = 100$, $\nu = 50$, $\mu = 40$, $\zeta = 0.3$, $\rho = 0.2$. The parameter setting follows a similar logic as the simulations in chapter 3. χ is unreachable, because limited cognitive capacities impose restriction in this exploratory simulation. The other parameters are set in a way that allows the computation of action and state values from reasonably large samples (ν, μ), and the revisiting of generalised nodes(ζ), since the environment is very dynamic.

Using the measures D, S and fit, figure 4.6 shows the networks obtained with this method. In the low cost range, density is 0.62, similar to the average base network model result. The same holds for the other cost ranges - density is 0.35 in the medium, and 0.04 in the high cost range. Thus, BRA generates the same outcome as the base model. In general, stability is lower than in the base model due to the increased amount of experimentation.

Figures 4.7 to 4.9 show the rules that were created during the process and how often they were activated. As useless rules are deleted by BRA, these appear with a lower frequency, rules that survived longer have a high frequency.

As all three figures show, there was no value in developing finer grained rules during the process. This is no surprise for the low or high cost range - in these scenarios utility is always increasing or mostly negative independent of the current state of the network. In the medium cost range, more rule experimentation is happening. For example, 36 mappings were generated

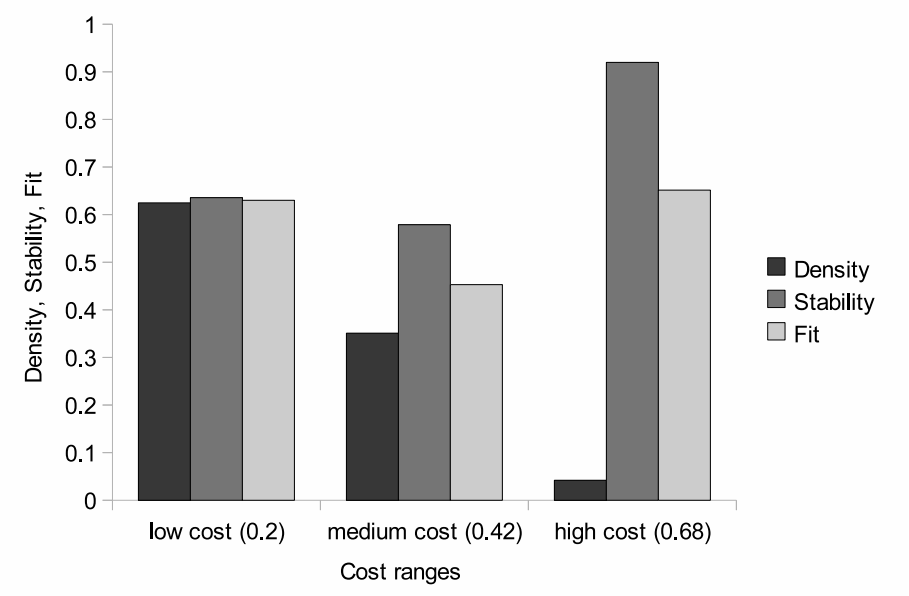


Figure 4.6: Network density, stability and fit for the BRA version of the model.

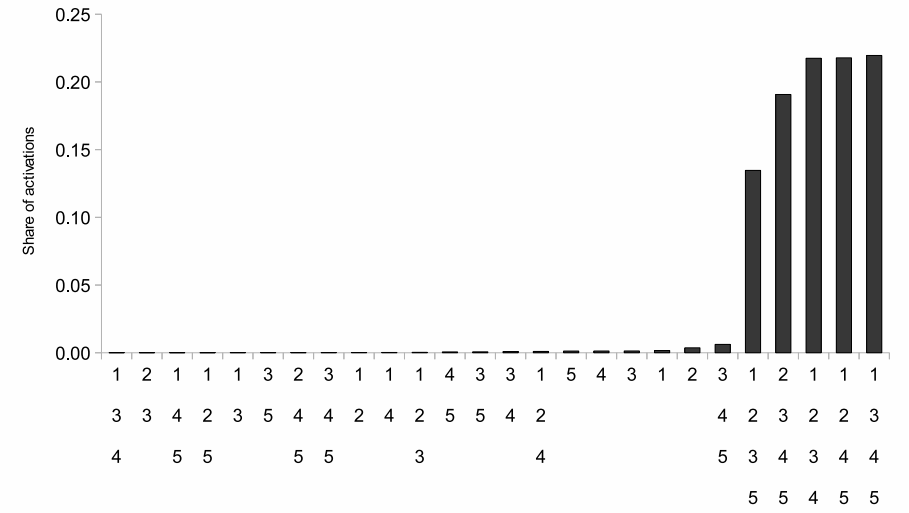


Figure 4.7: Rule extractions in the BRA network model for the low cost range. The labels denote the mappings, e.g. 1 2 3 4 represents the condition (player-name=1 *or* player-name=2 *or* player-name=3 *or* player-name=4).

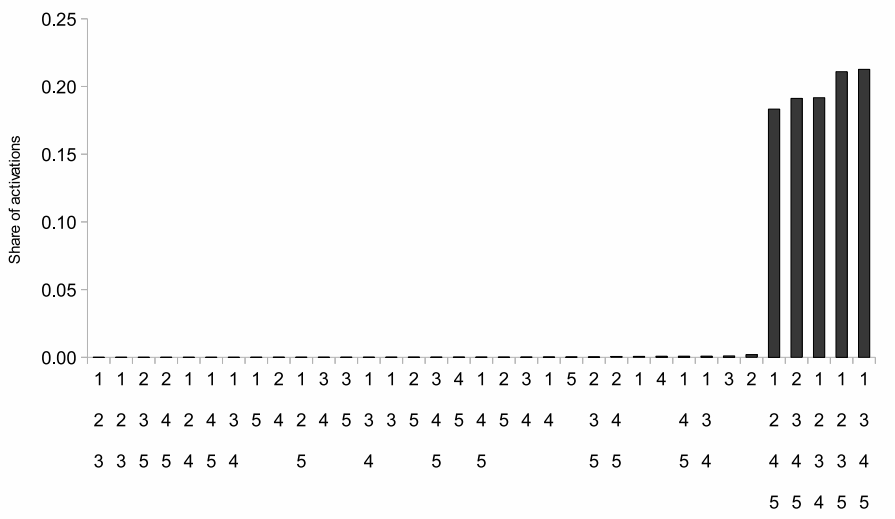


Figure 4.8: Rule extractions in the BRA network model for the medium cost range. The labels denote the mappings, e.g. 1 2 3 4 represents the condition (player-name=1 *or* player-name=2 *or* player-name=3 *or* player-name=4).

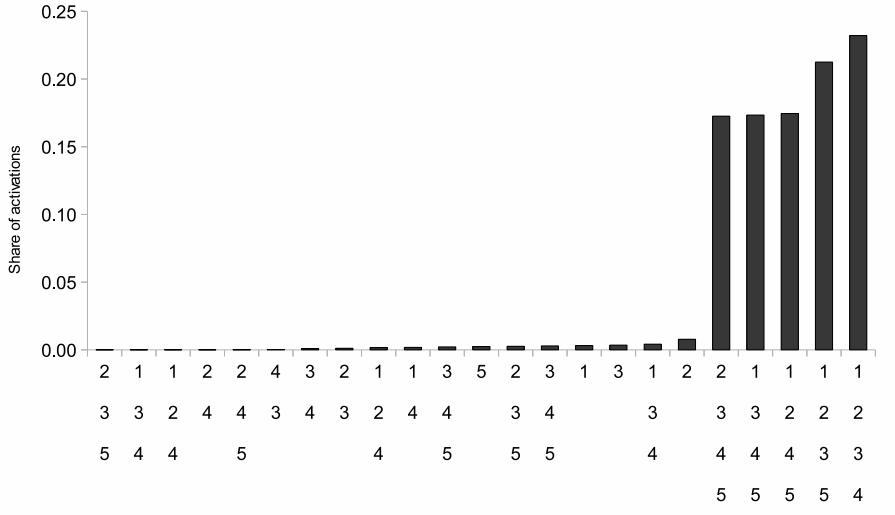


Figure 4.9: Rule extractions in the BRA network model for the high cost range. The labels denote the mappings, e.g. 1 2 3 4 represents the condition (player-name=1 *or* player-name=2 *or* player-name=3 *or* player-name=4).

as compared to 27 in the low cost range. The low frequencies, however, show that this does not lead to any sustainable mappings.

From this result, it can be concluded that the outcome in the medium cost range is similarly generated by a simple decrease in the selection probability, irrespective of the players who meet. It explains why the stability and fitness values are worse in the medium cost range; and it also shows that no specific model is necessary to generate the result. Simply decreasing the chance of offering a link is enough - at the price of higher instability.

4.8 Comparison with Empirical Results

After describing the structure of the networks resulting from the RL model, this section asks whether the presented RL model can explain empirical networks better than the equilibrium prediction.

However, with existing empirical results, comparison is not straightforward. As described in section 4.4, results in experimental game theory vary considerably. In the BG models, Nash networks emerge with a frequency of 0% to up to 40%. An evaluation of how well the RL model performs based on this data is difficult. In particular, except Vanin (2002) there is no experiment of the JW model. This model was, however, a first exploration where the co-operative nature of the game was investigated, but little quantitative data produced.

To gain some intuition how well the RL model does in predicting actual outcomes, here the experiment of Conte et al (2009) is simulated: The model is closest to a JW-type model as it requires mutual consent to establish and maintain a link. The following modifications were made to the RL model: δ is set to 1, i.e., there is no decay. All agents act simultaneously, so that all

possible parings happen at the same time step, so each agent has to make n-1 choices each round. This leads to much higher variability in the game, as from a single agent's perspective, the environment changes much more erratic as if only two agents moved at a time.

The remaining parameters are set as in Conte et al (2009)'s experiment described above. Table 4.6 summarises the parameter settings.

cost	benefit	α	δ	γ
90	100	0.01 ... 0.25	1.0	0.1, 0.25, 0.75, 1

Table 4.6: Adaptive network model parameter settings for the simultaneous linking game.

Simulations were run for α values up to 0.25 and some γ values. Each simulation is run for 100 steps and repeated 10 times. This is longer than the original 20 rounds, but was chosen deliberately to gain more representative results (whereas the variation in the experimental results is high due to the small numbers). To compare the result to the original model, the average payoff (over all simulations and time steps) is used. In Conte et al (2009), this value is given as 175.056 (standard error 7.901). The most similar values fall into simulations with $\gamma = 0.75$ (see figure 4.10).

From these runs, the simulation with the smallest difference from the experimental result in average profits and standard variation is selected *. This turns out to be the setting $\gamma = 0.75$ and $\alpha = 0.19$. Table 4.7 compares the average payoff from the experiments with the payoff resulting from the theoretically derived Nash equilibrium and the simulated results. While also the simulations do not match perfectly, they are with an expected value of

*This was the only aggregated figure available at the time of writing. It was not possible to obtain the results from the authors as their paper was under review at that time.

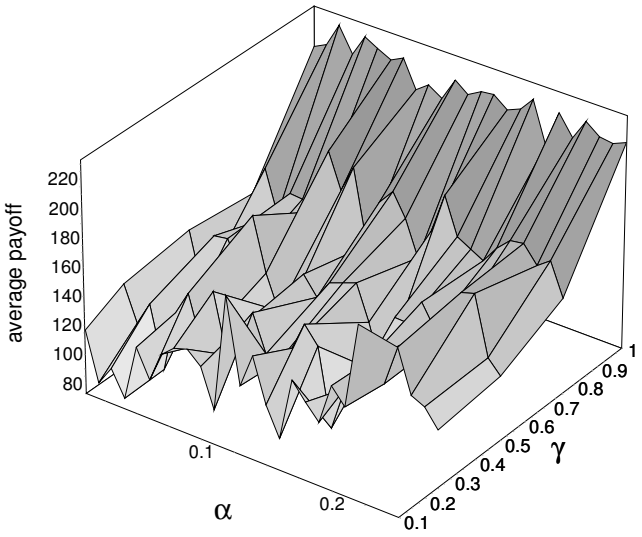


Figure 4.10: Average payoff for various α and γ values in Conte et al (2009)'s simultaneous linking game.

about 172 much closer to the actual result than the equilibrium prediction (a line network) with 296.67.

$\text{Payoff}_{experimental}$ (s.d.)	$\text{Payoff}_{simulated}$ (s.d.)	Payoff_{nash}
175.056 (86.55)	171.62 (125.45)	296.97

Table 4.7: Comparison of payoffs of equilibrium prediction, experimental and simulated results in the simultaneous linking game(γ=0.75, $\alpha = 0.19$, 10 repetitions

.

Figure 4.11 illustrates the dynamic of the simulation using a measure of stability, fitness and density for illustration. Stability is defined as above in definition 15, that is, as the likelihood that an agent changes a link. The share of Nash networks indicates how often the agents formed Nash networks (i.e. minimal connected networks) in the simulations. Although the simulations achieve quickly their final state with a Nash frequency of up to about 20% (average: 14%), it is also obvious that stability is not

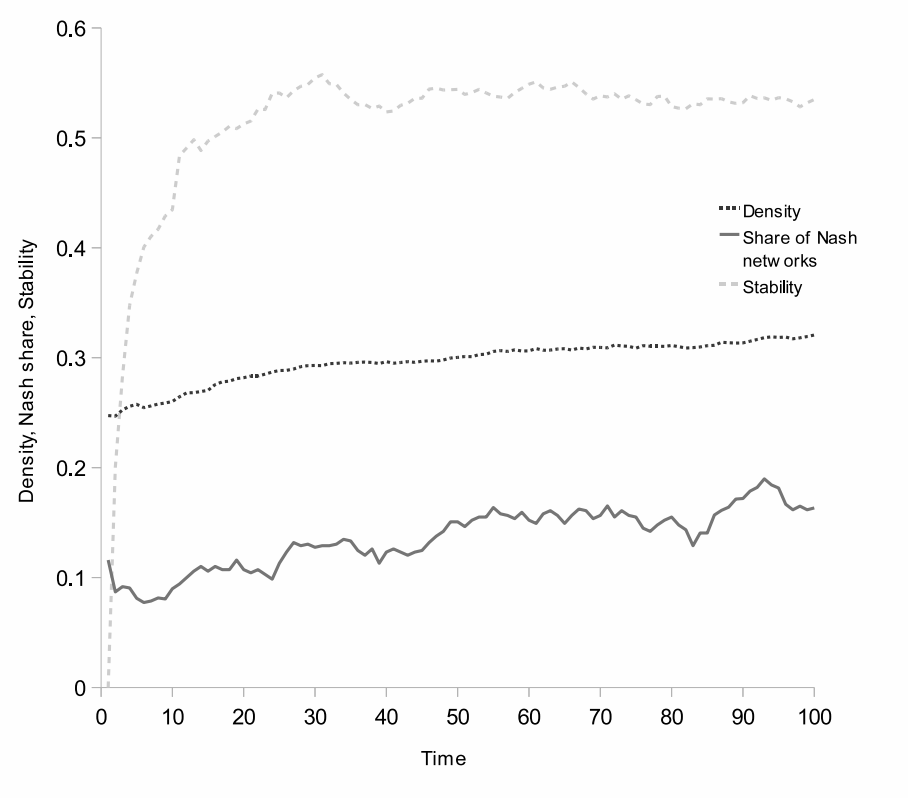


Figure 4.11: Density, stability and frequency of Nash networks over time in the simultaneous linking game ($\alpha = 0.19$, $\gamma = 0.75$, 10 repetitions). Values are computed as moving averages over 10 time steps.

very high. A value of only slightly about 0.6 means that almost every second agent chooses a different strategy each time step. The share of Nash networks increases slightly the longer the simulation runs.

Table 4.8 shows the Nash networks that emerged. The most frequent network is 1,1,1,2,2,3 with a share of 0.078. The efficient star occurs only three times during the simulations. Moreover, the most frequent network was the unconnected network 0,1,1,2,2,2, which appeared only slightly more often (share: 0.08) than 1,1,1,2,2,3. Thus, similar as Falk and Kosfeld (2003) observe, if a network is connected, there is a high chance that it is a Nash network.

pattern	stability	share	count
1,1,1,1,1,5	0.71	0.0004	3
1,1,1,1,3,3	0.56	0.0086	58
1,1,1,1,2,4	0.65	0.0108	73
1,1,2,2,2,2	0.55	0.0422	286
1,1,1,2,2,3	0.54	0.0771	522

Table 4.8: Nash networks visited in the simultaneous linking game. *share* represents the share of the network of all networks visited during the simulations. The number of total observations is 6773.

Although no exact comparison between the RL network model and the empirical studies are possible, the following parallels and differences between the RL model and actual human behaviour emerge:

- Nash networks are a good predictor for outcomes of the network game. It is not necessarily myopic, rational behaviour that causes this result. The frequency in the RL model is, however, low (about 15%). Many experiments of BG models report similar figures, but variation is high (from 0% to 40%).

- The RL model matches the empirical outcome (measured by the average payoff) much closer than the static equilibrium prediction.

- The RL model is very unstable. This holds, to some extent, also for the empirical results; some authors report a tendency to experiment after a stable solution emerged for some time steps. Many experiments never converged to a stable state. However, variation is lower, as for example observed in Conte et al (2009). There, in one instance convergence was observed. Following most authors, this is due to the tendency of real players to behave strategically. So, Callander and Plott (2005) find that some subjects take into account future

outcomes, which is of course impossible to capture with a simple RL model.

4.9 Conclusion

In this chapter, a reinforcement-learning version of Jackson/Wolinsky's connections model was presented and studied with simulations. The simulation results have been compared with the equilibrium predictions using the concept of stochastic stability as developed in Jackson and Watts (2002). The patterns (high connectivity in the low, medium connectivity in the medium, and low connectivity in the low cost ranges) are similar, but that there is some considerable distance between the equilibrium and RL model predictions.

The outcome of the RL process is driven by marginal utility, which has very different forms depending on the cost range. In the low range utility is convex but always positive; in the medium cost range, it slopes downwards after a certain density of the network is reached; in the high cost range, it is strictly negative. For a probabilistic choice model, this results in random switching in the low cost range the more connected the network becomes; low rates of experimentation in the medium range once utility starts to decrease; and punishment of any links in the high cost range. Moreover, the emerging structures in the medium ranges are most likely to be ring-like, as was stated by other authors like Watts (2001) or Bala and Goyal (2000); but this does not correspond to the outcome that was computed based on the stochastic stability approach in Jackson and Watts (2002), which resulted in a star.

Simulations with the BRA approach (chapter 2) showed that the same results can be generated with a simpler rule. The outcome of the algo-

rithm was that agents did not differentiate between players, but apply the same linking probability to any player they meet. This suggests that the more elaborated mental model of section 4.5 (remembering each player separately) does not add anything to an agent's utility. This is similar to results stated for two-player games in the experimental game theory literature (e.g Erev and Roth 1998).

In behavioural game theory, experiments with network formation are mostly based on non-cooperative network formation. Equilibrium outcomes with homogeneous agents are difficult to obtain with human subjects. At most about 40% of experiments converge to equilibrium. To find out whether the RL model predicts actual human behaviour better than the equilibrium prediction, another set of simulations with a modified setup based on the experiment of Conte et al (2009) was conducted. In the simulations, about 15% of the emerging networks were Nash. Using payoffs as a criterion for comparison, the RL model predicts much better. However, the stability of the simulated networks is lower than in the experiments. Moreover, in the RL model as well as in some of the reviewed literature, most connected networks were Nash. That is, this equilibrium concept describes empirical results well if the network becomes connected. However, it does not reveal anything about its frequency. This is estimated more accurately by the RL model.

Concluding, simple RL can be seen as a better predictor for actual human behaviour in network formation situations than the equilibrium prediction. It reproduces both theoretical patterns (although not to the same degree) as well as empirical phenomena. Thus, the RL network formation model contributes by adding an experience-based learning approach, which is situated between both strands of the literature. It provides a possibility to find out how likely a theoretic prediction is; while Nash equilibrium is a

useful concept for the type of result to be expected, the RL approach is a useful way to estimate the chance that this occurs in reality.

Chapter 5

The Market for Primary Care

5.1 Introduction

Health Economics typically treats health systems as linear systems that can be tested with statistical tools. Often, however, reality is more complex. Heavy interventions may cause only small changes, or compromise policy goals in different dimensions. In Kernick (2006)'s view, one could also characterise health care as a complex system, and argue that the construction of linear models leads to the omitting of system elements that are in the end the driving factors for the response of the system to policy interventions based on these linearised models. In complex systems, heavy interventions may have negligible influence, or small interventions may have a large effect. Interactions on different levels might produce unanticipated consequences, because on the macro-level these interactions cannot be accurately modelled. For example, a reform that allows patients a choice of health providers might remain without consequences if the doctors are reluctant to support their patients' decisions because they, say, see their influence and prestige in danger. The system can remain in an unchanged

state. Other interventions might change the system only in the short run because other factors restore the original state. For instance, higher patient mobility might first reduce waiting lists as patients search and so distribute more evenly across providers. However, in the long run waiting times might increase again, because, for example, some providers become highly sought after due to their reputation, while others are underemployed.

On the more specific level of general practice patterns, Scott (2000) points out that on the micro-level not much is known about doctors' decision making. There are, however, other, non-economic factors important for decision making. For example, doctors refer their patients to specialists out of uncertainty, or follow the pressure of their patients for certain treatments or prescriptions. However, economic models so far have not considered the possibility of such interactions on the individual level and their consequences on the aggregate level.

To find out whether complex systems theory provides an answer to the limitations of current research methodologies is, according to Kernick, a matter of years. In his view, a research program is needed that encourages the development of new statistical tools, experimental work to support theoretical constructs and demonstrate their usefulness, tools that promote systematical thinking about healthcare and a more widespread application of models that encourage dialogue between the stakeholders in the health economy.

The purpose of this chapter is to develop an agent-based model of primary care and to add a computational model to such a research agenda. A distinctive feature is the modelling of different assumptions about consumer behaviour on the individual level. Consumer behaviour in general has often been described as routine or habitual behaviour. This fits a special case

in the BRA learning framework presented in chapter 2. RL will here be applied to model patient choice of the general practitioner (GP). If a consumer knows n doctors, each doctor can be represented as a choice or action alternative. Using definition 4 this case is represented in BRA by: $k = 1$, $|A^k| = n$, and $succ(\mathcal{L}_0^k) = \emptyset$, with $A = \{choose(GP_1) \dots choose(GP_n)\}$. k represents here the situation that a consumer is feeling ill, i.e. becomes a patient. Thus, patients are faced with a single condition (being ill), under which they choose among different GP alternatives.

The plan of this chapter is as follows: In the next section 5.2, the health economic background is briefly outlined, before describing GP and patient behaviour in some more detail in sections 5.3 and 5.4 based on the available literature. Section 5.6 specifies the RL model. Section 5.7 presents simulations. First, the model is simulated with a large range of parameters to gain more understanding of its overall behaviour. Then, section 5.7.2 provides more detailed, dynamic results.

5.2 Background

The efficient provision of health care and its quality are central objectives of government policy. Especially in gate-keeping systems as in the UK, 'general (or family) practice and its role is increasingly regarded as the key to achieving efficiency and equity in many health care systems', as Scott (2000) notes. GPs influence the total cost of health provision; for example, they generate direct costs by referring patients to secondary care or prescribing medication. More indirectly, GPs may influence health costs by raising the health standard in general, e.g. by supporting preventative care.

To influence the way health care is delivered, primary care can be either

managed and controlled directly by, e.g., employing GPs as salaried personnel; or indirectly by setting financial and other incentives for self-employed practitioners. Direct control is difficult to achieve because it is expensive and difficult to implement, and because professional organisations try to preserve the independence of their members. Only in recent years, with the advance of information technology has performance-based pay become more common. The typical and by far most important approach is, however, to set financial incentives and modify political and organisational constraints. The function of designing incentive systems has been described as a way to align the government's objectives with the physician's interests, and implies that governments as principals may have different interests than health care providers. The common assumption is that GPs are income maximisers, an objective which may conflict with the efficient provision of health care (e.g. by providing more services than necessary). Consequently, the design of such systems is closely related to the principal-agent problem. Since in the majority of countries with public health policies the main instrument to shape the way health services are delivered is their reimbursement, most attention has been paid to the setting of financial incentives. Furthermore, many empirical studies find evidence that GPs do react to financial incentives. Another dimension of shaping GP behaviour, which has received more interest recently, is the promotion of patient choice. Here, the idea is to increase competition among GPs by increasing patient mobility. Where health providers are able to set prices, this may lead to increased cost-efficiency and/or quality; where prices are regulated, competition can motivate GPs to provide better quality services in order to attract and bind their patients.

Scott (2000) points out that to understand and judge policy interventions better, more attention has to be paid to the context of GP decisions. Factors such as patience's health status are important variables in doctors'

decisions; doctors might be pressed by some patients to refer them and so on. Most principal-agent and econometric models tend to neglect such factors, and for the sake of analytical clarity or lack of data treat them as a residual category. The ACE model presented in this chapter will try to better account for these contextual factors by using its own concept of 'appropriate treatment' that is assumed to be an important decision variable of doctors.

5.3 GP Behaviour

A central problem in designing incentive systems in health care is informational asymmetry. The patient is no health expert and has to trust that the GP acts in his or her best interest. This increases the discretionary power of the GP (Grignon et al 2002). The GP has also an information advantage over the public insurer or government, e.g. with respect to the expected case mix, the necessity of certain treatments, prescriptions and so on. This makes it difficult to monitor and control behaviour directly or indirectly. However, although information asymmetry points towards problems of moral hazard, there are characteristic differences to a principal-agent relationship: Health outcomes are difficult to measure; usually not the patient pays for the service, but a third party; the utility functions of patient and doctor are, to some extent, interdependent (Mooney and Ryan 1993) - an important deviation from classical agency theory, which assumes independence of utility functions (Ryan 1994). Another often mentioned factor inhibiting moral hazard is the trust characteristic of the relationship. The doctor-patient relationship is usually long-term, in which patients invest trust. For the GP, trust is capital, and he or she has an incentive to maintain it by avoiding obvious profit-maximising behaviour and safeguarding the interests of the patient. If the patient gets a feeling of too many unnecessary treatments

or consultations, he or she may lose trust and search for a new doctor (e.g Scott 2000; Arrow 1963).

Despite these constraining factors, the economic literature typically focuses on the principal-agent nature of the GP-patient and GP-regulator relationship. Under the assumption of self-interest and opportunistic behaviour, the question becomes which incentive system encourages the GP to behave in the best interest of the patient (welfare and quality), as well as the interest of the regulator (cost efficiency, patient welfare, and quality). Several authors have analysed models of health care provision in the context of a principal-agent problem (e.g. Marinoso and Jelovac 2003; Zweifel et al 2005; Scott 2005; Jelovac 2001; Ma 1994; Chalkley and Malcomson 1998a;b; McGuire and Rickman 1999), of which the most relevant will be shortly reviewed here.

Numerous econometric studies building on assumptions posited by the principal-agent literature have been conducted to test hypotheses about how GPs react to financial incentives. The main results of these studies are also summarised.

Analytical approaches

Considering a health authority maximising patient welfare minus expected cost, Zweifel et al (2005) analyse optimal contracts. The provider utility function can be written as $u(P, e) = P - C(e) - V(e)$. P is the pay, C are expected costs. The parameter e measures the effort to reduce these costs, and $V(e)$ represents the loss in utility due to these efforts. The payment can be expressed as $P = G + np + \gamma K$, where G is a basic allowance, a per capita payment p for n patients plus a share γ of the total costs K (i.e. service payment). At the one extreme, a prospective payment (capitation)

system is described by setting γ to zero. In this case, the provider bears all the risk. At the other extreme, a retrospective payment (fee-for-service, FFS) system is described by setting p and G to zero, so that the insurer bears all the risk.

A contract should internalise the principal's interests in cost effectiveness. At the same time, the contract must still be attractive enough to be accepted by the service provider. Following Zweifel et al (2005), the first-best solution FB is a payment system $E(P)$ that compensates the provider's costs, efforts to reduce costs and a reservation utility which the provider would achieve by not accepting the contract. This can be formally written as $E(P) = C[e_{FB}] + V[e_{FB}] + u$.

Varying the base model, they derive the following three typical cases with respect to cost efficiency: In case (1), the reference model, the provider is risk neutral, information is full and symmetric, and cost efficiency is the only objective. The first-best payment is given by $E(P) = C[e_{FB}] + V[e_{FB}] + u$, where u is the reservation utility that needs to be fulfilled for the provider to accept the contract. For this objective, a prospective payment system is optimal; more specifically, a lump-sum payment with which the provider has to cover all costs. The insurer can set the base payment in such a way that it covers expected costs. In case (2), GPs are risk-adverse, and the insurer has to pay a risk premium. It is then more effective to take over some of the costs to reduce the risk premium. However, this also reduces the incentive for the provider to reduce costs. Case (3) assumes that the provider has more information about the expected case mix, and thus over expected costs. An (opportunistic) provider will claim that he has only the most costly case mix to obtain a higher risk premium. By increasing payment with costs, the provider would be encouraged to share accurate information. This again reduces the provider's effort to reduce costs.

Three more cases can be derived when quality is added. Quality can be defined as the treatment success and the welfare of the patient. Treatment success may sometimes be observable. However, apart from measurement problems, it is impossible to determine whether a provider did not try to provide the necessary quality even if treatment was not successful. Extending the provider's utility function by assuming that quality has a utility V for the provider and that it comes at a certain cost C depending on the effort e the form becomes $U = E(P) - C(q, e) - V(q, e)$. Analysing this utility function, they find that a provider has no incentive to provide optimal quality: Case (1) is given by the assumption that treatment success and quality are observable and providers are risk-neutral. Then, an adjusted base payment induces the provision of optimal quality, as long as the reservation utility of the GP is met (i.e. basically, the insurer pays for the desired level of quality). If treatment success is stochastic and the more risk-adverse the provider is, the regulator has to pay a risk premium. In this case, direct control of quality is the cheaper option. In case (2), treatment success and quality are not observable. There is a trade-off between quality and quantity: For full take-over of costs in a pure retrospective system, the provider has no incentive to minimise costs; hence, he can raise quality until his marginal utility of quality equals the marginal cost of raising quality. Since providers in prospective systems have incentives to reduce costs as much as possible, quality will be minimal. In case (3), the regulator cannot judge success and quality, but patients can. If providers compete for patients, then capitation payment is the best option. In this case, there is an incentive to attract patients by improving quality, while at the same time to minimise costs. If the situation is monopolistic or the elasticity of demand is low, again a mix of capitation and fee systems is the best solution.

In what follows, some more specific models of primary care are reviewed,

focusing especially on the role patient choice plays: The models of Jelovac (2001) and Marinoso and Jelovac (2003) look in more detail at GPs' clinical decisions, and how these may be influenced by prospective and retrospective payment systems. Levaggi and Rochaix (2007) extend this model and explicitly look at the role of consumer choice in this setting. The models of Gravelle and Masiero (2000) and Karlsson (2007) treat capitation systems when patients choose between GPs.

In Jelovac (2001), patients can have a minor or a major illness. The minor illness can be treated by the GP; the major illness must be referred to a specialist. The GP must first diagnose the condition. The lower his effort, the less accurate the diagnosis; as a consequence of low effort, the patient may be mistreated. In case that the special illness was diagnosed but the patient had the general illness, the patient is cured, but with an unnecessary, expensive treatment. In the case that the patient was diagnosed with the general illness and is treated by the GP, but had the special illness, the patient was not treated accurately and has to be treated a second time. The doctor incurs a utility loss by mistreatment because a second visit is assumed to be costly, and because higher costs are incurred by the unnecessary treatment. In this model, capitation payment induces the most adequate treatment, since GPs are interested in decreasing the probability of a second visit and in minimising the total number of treatments. As a side-effect it induces higher effort as this is the precondition for appropriate treatment decisions.

Building on the same model setup, Marinoso and Jelovac (2003) provide some more conditions when prospective payment is more efficient than retrospective. They analyse three different strategies available to the GP: He may refer or treat blindly and save the effort of diagnosing; he can diagnose with a certain effort and then either treat or refer based on the outcome

of the diagnosis; finally, he can, under the assumption of asymmetric information, treat or refer irrespective of the diagnosis outcome. Since some cost and effort is incurred for accurate treatment, it only pays for the GP to diagnose accurately if the expected income is high enough. Otherwise, it is more rational to guess based on the expected case mix, and receive the net payment with the respective probability that the guess was correct. Jelovac and Marinoso argue that the right incentive system depends on the insurer's objective: If welfare loss (caused by inadequate treatment) is high, the most efficient option is to set incentives in form of treatment success related fee payment. However, if the welfare loss by inadequate treatment is not the most important objective, capitation payment is sufficient, as it induces the GPs to reduce efforts, including the diagnosis effort.

Gravelle and Masiero (2000) present a game to research the question whether increasing the capitation rate can induce higher quality. The model is a two-stage hotelling game with two doctors, and n patients. Doctors choose a level of quality; the higher quality the more costs are incurred by the practice. The patients' utility function includes distance and expected quality (which is unknown to the patients initially). In the first round of the game, patients choose a doctor based on their utility function. In the second round of the game, quality is revealed, and patients compare their expectations with the actual quality. Patients may then switch to the other GP in the second round. If they switch, they incur some switching cost. Gravelle and Masiero find that higher capitation rates increase quality as it makes patients more valuable to practices, even if patients care much about distance. They also show that GPs have incentives to increase quality even if patients misjudge quality. As a result, both doctors increase quality as long as costs are covered.

Karlsson (2007) develops a similar hotelling game. As in the preceding

model, patients choose in a first stage their GP based on distance and expected quality; after that, actual quality is revealed, and patients may switch to the other GP. Karlsson considers additionally the search behaviour of patients. Because of the interaction effects between consumer search patterns and provider reactions, there may be settings where the optimal capitation rate is indeterminate. If costs are very low, all providers have strong incentives to increase quality. The more GPs do so, the stronger decreases the variation in the GP population. This discourages patients from searching, since there is not much to gain from (costly) search in a homogeneous GP population. As a result, the equilibrium quality may decrease even with increasing payments, because patients have no reason to change providers. However, this happens only with a quadratic cost function, and the author assumes that hyperbolic cost functions are more intuitive and likely. Such cost function lead to an equilibrium where quality increases with the capitation rate.

Levaggi and Rochaix (Levaggi and Rochaix 2007) combine the patient choice perspective of Gravelle/Masiero and Karlsson with the moral hazard perspective of Jelovac and Marinoso. The setup is as in Jelovac and Marinoso (2003), but additionally, patients may choose the access route to either GP or specialist themselves. Thus, GPs as well as patients can make mistakes in a treatment choice. They find that under perfect information (where the severity of illness can be judged ex-post) a gate-keeping system is efficient. Its efficiency can furthermore be increased by allowing patients to seek specialist care themselves, provided patients bear some of the risk in form of payments for mistakes. If information is imperfect and opportunistic behaviour possible, then non-gate-keeping systems are more effective. Intuitively, this is because under capitation, GPs refer also mild illnesses; under FFS, GPs will first treat themselves, even if the condition is severe;

specialists will always treat rather than sending the patient back to the GP. So the patient is the only actor who has an interest in the effective provision of care (e.g., because he or she wants to avoid unnecessary visits). Even if the patient makes mistakes in judgements the result is more cost-efficient.

The conclusion of this short survey is that pure capitation systems are desirable for cost containment, but are optimal only under very restrictive assumptions like risk neutrality of providers and information symmetry. Some costs should be taken over in form of FFS. This reduces the willingness to save costs, but also the risk premium that had to be paid otherwise. Takeover of costs may as well increase quality if there is no or little competition between providers. However, this will depend on the information available to the health authority. If there is competition for patients, the size of capitation payments can act as an incentive to improve quality. Gatekeeping systems are only efficient under perfect information. Patient choice can have a cost-reduction and welfare-increasing effect in non gate-keeping systems, especially if payment is by FFS.

There are further, newer incentive systems in primary health care, which are not considered here. For example, Pay for Performance combines aspects of managed systems with financial incentives by making payments dependent on treatment priorities, and conditions treated. If a certain target is reached, reimbursement decreases, acting against over-treatment and opportunistic diagnosing. This requires much closer monitoring, which more recently has become possible due to the increased availability and usage of new information technology.

Empirical approaches

Types of studies Several empirical studies in the last 20-30 years investigated the influence of different remuneration schemes on GP behaviour. Much of this literature (until about the year 2000) has been extensively reviewed (Scott and Hall 1995; Scott 2000; Gosden et al 2000). Most studies find a relationship between payment system and practice patterns. However, the validity of the results is often limited to special circumstances, as most of them are 'opportunistic' studies, taking advantage of data collected for other purposes. Studies where the payment scheme was changed, or new elements in the incentive system were introduced, were the most important studies to investigate the relationship between payment and GP behaviour. The most rigorous selection of studies was applied by Gosden et al (2000), who reviewed only studies based on control group comparisons (randomised control trials), time-series data or controlled before-and-after studies. The advantage of such designs is a better control of confounding variables. Scott and Hall (1995) also included cross-sectional studies, where it is more difficult to estimate the influence of, say, self-selection effects of GPs into certain payment schemes.

As most studies are described in the reviews, only the main results of the most influential studies, and some of the newer literature are summarised here. The major studies are the following:

The Krasnik study (Krasnik and Groenewegen 1992) compared two groups of GPs in Denmark. GPs in the Copenhagen area moved from capitation payment to a mixed capitation/FFS payment mode, while for the regional doctors, the mixed capitation/FFS had already been introduced. Data were collected six months before and at a 6-month and a 12-month period after the intervention, allowing the comparison of practice patterns

of the same GPs before and after the intervention.

The Hutchinson study (Hutchinson et al 1996) compared the referral patterns in Ontario, Canada, where FFS payment was changed to a mixed capitation/incentive-based payment. A control group remained in FFS; the intervention group received capitation payment. Furthermore, for each hospital day exceeding the mean hospitalisation rate, the practice had to bear a third of the hospitalisation cost. The authors compared in detail for different patient groups the changes in referrals to hospitals.

The Davidson study (Davidson et al 1992) compared two groups of doctors paid by Medicaid. The capitation group received a per capita payment, and some amount per service, and could keep any surpluses on savings (but had also to cover losses up to a certain extent). The FFS group received higher fees for certain services as compared to the control group (fees were about half the size).

The Hickson study (Hickson et al 1987) analysed the introduction of salary payment in a FFS system. The salaried doctors received a fixed income per month, the FFS doctors a fee for each visit. Both incomes were designed on historical consultation rates, thus roughly equal in height.

Main results With respect to referrals, evidence is mixed - some studies suggest an impact of the payment system, some do not. The Hutchinson study found that Canadian FFS doctors did not lower referrals to hospitals. Likewise, the Davidson study found that the number of specialist visits was greater in the FFS group than in the capitation group. On the other hand, the Krasnik study found a decrease in hospital and specialist referrals after 12 months for FFS doctors, while there was no significant change in the short-run (after six months). The Krasnik study also observed a fall

in prescription renewals after FFS was introduced. This was unexpected because extra fees for prescriptions were introduced, so that there was no reason to reduce their costs.

With respect to the number of GP consultations, most studies indicate a rationing of visits by capitation (and salaried) doctors as compared to FFS doctors. The Hickson study found a lower number of visits per enrolled patient in the salaried physician group as compared to the FFS group. This was partially due to FFS doctors scheduling partly unnecessary services, to some extent, due to too few visits by salaried doctors. Furthermore, the Davidson study found that the number of primary-care doctor visits was higher in the new FFS group than in the capitation group. Moreover, other studies (e.g. Kristiansen and Hjortdahl 1992; Kristiansen and Mooney 1993; Kristiansen and Holtedahl 1993) found that GPs paid by FFS are more likely to provide shorter consultations.

Also with respect to the intensity of services provision most - although not all - evidence points in the expected direction. FFS increases service production, which is typically interpreted as the realisation of income opportunities by doctors. The Krasnik study found a strong increase in curative and diagnostic services after the change to FFS. Similarly, Kristiansen et al. found that FFS doctors are more likely to order tests (Kristiansen and Hjortdahl 1992; Kristiansen and Mooney 1993; Kristiansen and Holtedahl 1993), a conclusion that is also reached by Devlin and Sisira (2008) in their analysis of doctors with a mainly fee-based income in Canada. However, for Norway, Grytten and Sorensen (2001) find that GPs paid by FFS did not increase service production as compared to salaried doctors. Comparing practices with different list sizes, i.e. different demand, they also showed that practices with short lists have no higher service production per consultation to compensate income loss compared to those practices with higher

demand (Grytten and Sorensen 2007).

Summarising, evidence is not always as theory predicts. Regarding referrals, evidence is mixed; regarding the number of consultations there seems to be a clear trend for rationing under capitation; with respect to provider-induced demand generated by FFS, before-after studies find evidence for, cross-sectional studies against additional demand. Especially when looking at the different conclusions of cross-sectional studies as compared to before-after studies (especially the Krasnik study), the question seems rather not to be *if* the payment mode influences behaviour, but how large and important the effect is when considering the health system as a whole. Other influences such as rural-urban location, working hours and so on could have an impact that reduces the influence of the payment as a single factor to insignificant levels.

5.4 Patient Behaviour

Whereas the insurer-doctor agency problem has been extensively studied, the patient-doctor relationship has attracted less attention. This relationship is characterised by information asymmetry - because patients are usually not good doctors, they have to trust their doctors and expect them to act for their benefit. Furthermore, it is often impossible to judge whether particular treatments are unnecessary or not, or whether a different doctor would have been more successful in treating a certain illness.

The common view has usually been that the patient is only interested in health, i.e. health status is the only variable to his utility function. There are, however, other dimensions in the patient utility function. For example, patients might also expect some non-medical aspects such as a diagnosis to rule out a dangerous illness, or obtaining information before surgery (Ryan

1994; Mooney and Ryan 1993). Other authors have stressed the role of information and the involvement of the patient in the decision process for treatment options (summarised by Vick and Scott (1998)).

Empirically, discrete choice experiments about the patient-doctor relationship have been conducted to find out patients' preferences. Vick and Scott (1998) derive from the literature the following dimensions as being important for patients: Being able to talk to the doctor; information about the health problem; information about the treatment; doctor's information and explanation; who chooses the treatment; length of consultation; and waiting time. They find that 'being able to talk' was most important to patients, and that 'who chooses your treatment' was the least important. Waiting times seem to be of little importance when patients can see a doctor they know. Information about the condition and treatment were rated with similar importance in the middle. The dimensions in Hole's study (Hole 2008) were given by: Waiting time; cost (measured as willingness to pay); warm and friendly doctor; knowing the doctor; thoroughness of physical examination. In this study, reassurance about the process in the form of a thorough medical examination turned out to be the most important. These findings highlight also the 'non-functional' aspects of the doctor-patient relationship; whereas technically the patient has an illness to be fixed, the doctor additionally performs a social function by providing assurance or help in a general way. Thus, it might be that not necessarily the best doctor in clinical terms is preferred by patients, but maybe a doctor who spends more time with them and who gives patients a feeling of being taken care of. As Vick and Scott (1998) point out, convenience and accessibility factors such as opening hours or distance have already been found less important for the patient's utility function by other authors (e.g Williams and Calnan 1991).

Surveys using actual patient satisfaction survey data find similar results. Dixon and Robertson (2008) find that the quality of the relationship with their doctor is the most important factor influencing satisfaction, while the factors with the lowest predictive power are waiting time and accessibility. They find that once a good relationship is established loyalty is high. Practice change, they conclude, will probably only occur if the relationship breaks down, so that increased choice is not expected to increase patient movements significantly. This has also been observed before, although patient choice was not high on the agenda then. Low mobility has been attributed to unfavourable circumstances preventing dissatisfied patients from changing their GP (Gage and Rickman 2000; Goodwin 1998; Gabbott and Hogg 1993).

Still, the impact of patient choice on the primary care system is mostly unknown. In the UK, for example, most evidence used to argue in favour of choice stems from pilot studies in secondary care. Critical authors (e.g. Appleby and Dixon 2004) state that most arguments in favour of choice remain rhetorical as no facts nor clear conditions are given. So, for example, patients in the London choice pilot studies were only allowed to choose if waiting time exceeded a maximum; as a result average waiting times decrease inevitably simply by design of the study. The impact of choice on quality was not measured at all. It has also been argued that actual quality improvements are less due to the switching of providers, but rather the image concerns of hospital managers (Robertson and Thorlby 2010).

Summarising, relatively much is known about stated preferences of consumers in health care; very little is known about actual choice behaviour or whether increased possibilities of patient choice will have an impact on the efficiency (e.g. reduced waiting times) or quality of primary care services. The available surveys as well as pilot studies from secondary care simply do

not provide suitable data.

5.5 Modelling Primary Care

The preceding sections revealed that existing knowledge about the driving forces of efficient health care provision has some shortcomings, which are mainly the following:

- The agency literature makes strong assumptions about doctors' motivation. Many analyses treat in detail only the extreme case in which doctors are not interested in their patients' welfare, or constrained by professional standards. The few articles accounting for joint patient and GP utility functions remain vague.
- The empirical literature is constrained by the data available. Only a few studies had the opportunities to study explicitly the influence of different remuneration systems in longitudinal designs. Not surprisingly, some results remain inconclusive. For example, some studies find that capitation payment leads to more referrals, some find the opposite.
- With respect to quality, there is a theoretical consensus that competition is likely to improve quality, especially when prices are administered (Gaynor 2006). However, the extent to which patients are willing to change providers is unknown. Furthermore, nothing is known about the implications for the health system as a whole. So far, data has mostly been collected in exceptional circumstances, as for example during the (secondary care) patient choice pilot studies.

A complete model of primary care can certainly not replace missing data, but at least simulate scenarios and highlight possible policy impacts, which in the current discussion remain purely theoretical or politically motivated. The computational model developed in the next sections will look mainly at the dimensions of quantity and quality of service provision. The following hypotheses inferred from the literature will be the frame for these scenarios:

- Prospective (capitation) payment is likely to ration service quality and quantity. If patient demand reflects quality, and there is competition between GPs, quality may rise if the marginal capitation payment exceeds the marginal cost (strong empirical evidence from most studies).
- Prospective payment is likely to induce higher than necessary rates of referral. This effect might be counterbalanced if effort is high. In this case, GPs want to attract more patients by better services; better services could be reflected by more appropriate treatment (no empirical evidence; the hypothesis is mainly based on Karlsson (2007) and Gravelle and Masiero (2000)).
- Retrospective (FFS) payment induces in general a higher volume of services (supplier-induced demand hypothesis; empirical evidence from most studies). As competition increases, GPs are likely to increase unnecessary treatments to compensate for short lists (based mainly on the supplier induced demand hypothesis (Zweifel et al 2005); there is only weak or even no empirical evidence (e.g. Grytten and Sorensen 2007)).
- As retrospective payment induces a high provision of services by the doctor himself, the rate of referrals is expected to be lower than in capitation systems, i.e. there are no unnecessary referrals. In fact,

there may be fewer referrals than necessary (some empirical evidence from studies).

– Demand side induced competition by patient choice improves quality. Unsatisfied consumers are likely to change their GP, but then are likely to remain loyal, decreasing competition again. Under FFS, this can be expected to work against unnecessary treatments (to prevent excessive exits of existing patients). Under capitation, patient induced competition will reduce unnecessary referrals to attract and keep patients.

The computational model of primary care is described in the next section. The aim is to design it in a way that allows to investigate the hypotheses that so far have only been incompletely covered in existing work.

5.6 A Reinforcement Learning Model Of Primary Care

The actors in the model are patients, GPs and the health authority (HA). The HA is setup once per simulation. Its main function is to implement the policy for a simulation run (e.g. by defining the value of fee and capitation payments), and to pay the GPs.

The main assumptions of this model are:

– GPs are self-employed professionals who trade leisure and patient welfare against income. Costs are incurred only indirectly in terms of effort and time the GP invests.

– GPs can define their maximum workload, which must be > 0.

- There is no reservation utility of GPs. A GP must treat patients coming to him.
- There are no exits of GPs.
- There are no switching costs. When patients become unsatisfied with a GP, they may search for a new doctor without incurring any transaction or search costs.

5.6.1 Overview

Patients and GPs are distributed randomly over a grid, with x and y dimensions from 0..1.

Time in the simulation proceeds in a discrete way. A time step d represents exactly one day. A period t is defined as a number of days. For example, a period could be a week ($t = 7$ or a month $t = 30$). Typically, certain decisions and updates are made per period, not per day.

At each day $d > 1$, consumers face a certain probability of becoming ill with condition $1...m$. When they become ill, they choose a GP based on their utility function (see section 5.6.2). If it is their first appointment with this doctor, the GP adds the patient to his list $list$. Then, an appointment is scheduled by the GP at time $d+k, 0 <= k$. k depends on the waiting list $list_{wait}$ of the GP. A GP can treat up to $appointments_{max}$ patients a day. As long as the waiting list for the day is not filled ($list_{wait} < appointments_{max}$), $k = 0$. After that the appointment is made for $d + 1$ and so on. k is thus the number of days until a patient is seen by the doctor.

A GP sees then up to $n, n < appointments_{max}$, patients per day. Depending on the condition a treatment is chosen. While the condition is always diagnosed correctly, the treatment choice is uncertain. This uncer-

tainty is represented as a probability with which a doctor chooses between alternatives 'treat or 'refer'. For example, some severe illnesses must be referred - the GP has to decide to refer with probability 1. There are other, milder conditions for which the GP can decide either to refer or to treat himself. The details how this choice is made is described in section 5.6.3.

After the consultation happened, the patients receive information about the doctor (e.g., the effort the GP made, or the waiting time to get an appointment), which then enters their decision-process at the next time they become ill.

The HA pays the GPs at the end of a period. The GPs send their bills, containing the services they provided as well as the patients on their list, to the HA. The HA calculates the pay depending on the policy being implemented and sends the amount back to the GP. The GP then updates his utility for the period and decides his work plan for the next period (e.g., the number of patients to see).

In the following sections, the utility functions and decision processes of GPs and patients are described in detail.

5.6.2 Patient Decisions

Utility function When patients become ill, they choose a GP and make an appointment. The choice of GP is based on distance $dist$, experienced waiting times $wait$, and experienced effort of the doctor, E. Effort is here interpreted as an indicator for the quality of the doctor-patient relationship, for example, the time the doctor spends per consultation.

A patient calculates his welfare by

$$U_P = \alpha wait + \beta dist - \gamma E. \tag{5.1}$$

To be precise, U_P is the 'inconvenience' of the consumer, which he tries to minimise (minimal distance, minimal waiting time, maximal effort of the doctor). Before knowing a doctor, the patient has no knowledge about waiting times and effort. The function then reduces to $U_P = \beta distance$.

Calculation of the utility function After the first experience, a patient can update his doctor information with respective values of E and $wait$. Each patient maintains a list of GPs. The model generates and updates this list; if the list is full, but there are more GPs unknown to the agent, the model may replace the worst ranking GP from the list with a new candidate.

Decision process Patients only consider practices if waiting time is <3 days. If all doctors a patient knows have a waiting time ≥ 3 days, he chooses the closest GP. Patients forget the actual waiting time by reducing the experienced waiting time by a certain factor each following day (currently set to $\frac{wait}{10}$). Thus, even if waiting time was long some time ago, a patient might consider visiting this doctor again. Forgetting is important in this model, because otherwise some practices would have no chance to convince dissatisfied patients to return, as patient decisions are based on experience.

Patient choice behaviour is modelled using different forms of rationality. A first dimension in which patients are bounded rational is in the sense that they have access to only a small amount of information. First, the number of doctors a consumer can remember is set to 3. Second, information about GPs is circulated in networks of consumers. The network size may vary; here networks with 2 or 5 close neighbours and one distant link (randomly

chosen from anywhere in the landscape) have been generated. The idea is that in a world with small networks, information circulates less freely. The distant link has the function to bridge the distance among all consumers in the landscape so that it is less likely, but not impossible to learn about the best choices in the model. A second dimension of limited rationality is realised by the application of different choice modes. In the 'rational' choice mode, a patient ranks all GPs by their expected utility, given by the last experience (i.e. there is no discounting) and chooses the one which ranks highest. That is, even infinitesimal small differences are recognised by the agent. This is what meant with 'rational' - the agent utilises its computational power to distinguish between smallest difference in utility. The 'probabilistic' choice mode is given by simple RL. In this case, GPs represent action alternatives, and the expected utilities are used as the payoff p for equation 2.9 in chapter 2 to update the action strengths, and thus the selection probability for choosing a particular GP. In the probabilistic decision mode, small differences between GPs will lead to similar choice probabilities.

Combining the behaviour dimensions - information availability and choice mode - will allow us later to relate patient behaviour to the findings from the literature review. For instance, it will help to investigate the difference between scenarios where consumers have access to more information (larger network), or make more efficient use of that information (by more rational decision making).

5.6.3 GP Decisions

Decision context and constraints In this model, doctors' decisions are not influenced only by their own welfare (income), but also by their patients' welfare and normative constraints like professional standards, which

prevents them from being purely selfish. So even if a GP has not reached his or her preferred income, he or she will not necessarily provide excessive treatment - because it is neither in the patient's best interest, nor is it justifiable before himself or herself or other colleagues. If there is, on the other hand, room for 'interpretation' whether additional treatment is necessary or not, his own welfare may play a larger role in deciding.

The model uses a decision-theoretic approach to reflect such situated decision processes. The central concept in this approach is the clinical condition and the related treatment(s). One patient can have exactly one condition m, for which exactly one treatment tr exists. This is labelled a condition-treatment pair $\{m, tr\}$. The diagnosis is always correct, and the doctor has only some discretion about whether to apply the treatment or not.

The condition-treatment pair $\{m, tr\}$ determines the likelihood with which a patient is treated by the GP or referred to secondary care. A decision is always, to a smaller or larger extent, uncertain. The GP's decision is therefore modelled probabilistically.

The condition-treatment pair specifies also the effort necessary to apply the treatment, which can be seen as a sort of cost accrued by the GP when choosing the option to treat. Referral has no effort for the GP - treating is always more 'costly' than referring.

There is only small variation in the probabilities of each outcome (referral or treatment) of a consultation. The upper and lower bounds of this variation are determined by parameter var_{max}, which sets the maximum deviation from an objective norm. The extent of the actual variation var_{actual} depends on the individual utility function and is adapted by the GP during his decision process (see below). In detail, the decision proba-

bilities are calculated as follows: Each $\{m, tr\}$ has a professional certainty value p between 0 and 1. This value indicates the certainty that tr is the 'appropriate' treatment for condition m; conversely, $1 - p$ represents the opposite. The idea is based on Krasnik's operationalisation of uncertainty (Krasnik and Groenewegen 1992): Krasnik measured professional uncertainty as the regression coefficient of treatment and condition in the GP population. For example, a coefficient of 0.24 for a condition-treatment pair (or better, condition-treatment group, because some services are applied typically for a number of diagnoses) means that 24 % of all doctors apply this treatment if they diagnose the respective condition. This can be interpreted as little professional consensus about whether to apply this service, as 76 % of doctors would do nothing, or refer. Without accounting for the empirical distribution of such certainties, the model uses this idea to define a 'norm' for each $\{m, tr\}$. So, for example, if $p = 0.24$, a single GP decides to treat a condition i with $p_{i,actual} = p \pm var_{i,actual}$, and with $1 - p_{i,actual}$ to refer.

Utility function The GP is seen here as a self-employed professional: The objective is to earn some income with minimal effort; at the same time, he cares for his patients' welfare. The utility function is given by

$$U_{gp} = I^{\alpha}(E_{max} - E)^{\beta}(n - DEV)^{\gamma}. \tag{5.2}$$

The rationale behind this Cobb-Douglas type utility function is to capture the decreasing marginal utility that is the usual standard form of utility functions. It is positively influenced by total income (I) per period, total 'leisure' ($E_{max} - E$) per period, and the welfare ($n - DEV$) of all n patients in a given period. Leisure is defined as the difference between maximum effort E_{max} and actual effort; the larger this difference (i.e. the smaller the actual effort E), the higher utility. A similar logic applies to the valuation

of patient welfare: Welfare is a function of the appropriate treatment, where appropriateness is defined as the minimum deviation of a single patient's treatment from the norm; DEV stands for the sum of these deviations. The more patients are over- or under-treated, the smaller the contribution to total utility ($n - DEV$ becomes smaller). The calculation of the variables I, E, E_{max} and dev is described in the next paragraph.

Calculation of the utility function The components of the utility functions are computed the following way:

Income is defined as

$$I = lI_{capitation} + \sum_{i=0}^{|TR|} q_i fee_{TR_i} \tag{5.3}$$

l denotes the list size of the GP, TR is the set of treatments a GP may apply, and q_i the number of times a particular service i from set TR, denoted TR_i, was actually applied. Payment is given by a capitation fee $I_{capitation}$ per patient, and the sum of fees of applied services.

For the calculation of leisure, the difference between the maximum possible effort E_{max} and actual effort E is calculated. The actual effort is given by

$$E = nE_{base} + \sum_{i=0}^{|TR|} q_i E_{TR_i} \tag{5.4}$$

Each service TR_i requires some effort (e.g. the time necessary to perform the service), which is the same for every GP. However, a GP may vary the 'base effort' E_{base}, $0 < E_{base} < 1$, per patient. This base effort stands for, e.g., the time spent with the patient, information and explanation given during a consultation and so on. Thus, the lower the effort per patient, and the lower the probability of treatment, the smaller the effort (the more

leisure) of the GP. E_{max}, the average maximum possible effort is calculated by the same formula, substituting the maximum possible values for the variables:

$$E_{base} = 1,$$

$$n = \sum_{i=0}^{|TR|} q_i = appointments_{max},$$

$$\bar{E}_{TR} = \frac{1}{|TR|} \sum_{i=0}^{|TR|} E_i(p_i + p_i var_{max}).$$

The average maximum effort $\bar{E}_{TR}$ over all possible treatments is given by the average maximum treatment probability p and the effort values E for all $|TR|$ treatments. Multiplying now $\bar{E}_{TR}$ by the maximum number of patients $appointments_{max}$ and the highest possible effort per patient E_{base} defines the maximum possible effort a GP can have per day: $E_{max} = n\bar{E}_{TR}$,

The measurement of patient welfare is not well defined in the literature and difficult to operationalise. Evans (1976) sees the over-provision of services constrained by some professional or ethical standard limiting the power of the income or leisure component's in the GP's utility. Lerner and Claxton (1996) point in their analysis of utility functions to authors with similar arguments: Dranove (1985) argues that too aggressive provision of services might lead to patients leaving or reduce the number of visits. Woodward and Warren-Boulton (1984) state that 'each physician derives additional utility both from positive consumption of the product of his leisure activities... and from providing additional care per patient ... up to the 'appropriate' amount'. Based on such arguments, it is assumed here that there is some norm of appropriate treatment, acknowledged by the professional community as well as by common-sense of patients. The assumption of the existence of such a standard allows the definition of appropriateness

as zero deviation from the norm. For each patient, the absolute difference of the consultation outcome with the professional certainty p is computed. So to speak, this is the deviation of what was objectively expected to be appropriate for the patient, and what the GP actually did. Thus, for all individual decisions with uncertainty (0 <p< 1), there might be a deviation. Since the modeller cannot judge the individual clinical decision, only the sum of all positive and negative deviations is taken into account for defining welfare:

$$DEV = \sum_{i=0}^{n} \sum_{j=0}^{TR} (decision_i - p_j) q_{i,j} \tag{5.5}$$

$q_{i,j}$ equals 1 if treatment TR_j was applied to patient i, 0 otherwise. $decision_i$ equals 1 if the patient i was treated, 0 if referred. Thus, for all patients of the GP, DEV approximates zero if the doctor treats always according to the norm (and if patient numbers are sufficiently large); values > 0 denote overtreatment (the larger the value the more likely treatment), values < 0 undertreatment (the smaller, the more likely referral). For example, if $p_j = 0.8$, i.e. professional consensus points strongly towards treating, and if $p_{j_{actual}} = 0.5$ (the doctor reduces effort by referring), then of 10 patients with that condition 5 are referred, and 5 treated: $DEV = 5 * (1 - 0.8) + 5 * (0 - 0.8) = -3$, whereas for a 'norm-conform' doctor it would calculate $DEV = 8 * (1 - 0.8) + 2 * (0 - 0.8) = 0$. Patient welfare $n - DEV$ for the first doctor would be 7, for the second 10.

Decision process The GP's decision concerns the setting of his treatment pattern p_{actual}: $p_{actual} = \{p_{0,actual} \dots p_{i,actual} \dots p_{z,actual}\}$ (determining the referral behaviour), the number of appointments per day n (determining the workload, respectively leisure $E_{max} - E$ and income I) and the consultation

pattern, which is the effort invested into the doctor-patient relationship E_{base}. The GP can influence income and leisure by setting the maximum number of appointments and the treatment pattern (if there is fee income); and he or she can influence patient welfare by setting the treatment pattern.

Several scenarios can be generated with this set of variables. For example, some doctors may prefer to set E_{base} low and see many patients with a high probability of referring them. This would be an income-maximising, effort-minimising strategy for a doctor without concerns for patient welfare. Furthermore, this strategy would pay best in environments with a large capitation component since there is no income loss from not treating. However, as patients can react via evaluation of E_{base}, and may switch to another doctor, the relationship of the GP's decision variables become non-deterministic.

The decisions of the GP in detail are as follows: At the end of each period the GP decides about his consultation pattern and how many patients n_t he wants to see each day in the current (beginning) period t. To do this, he 'simulates' the optimal configuration of the treatment probabilities p_{opt} for all treatments: $p_{opt} = \{p_{0,opt} \dots p_{i,opt} \dots p_{z,opt}\}$, and appointments n_{opt}, $appointments_{min} < n_{opt} < appointments_{max}$ for the next period t. He uses for this the known constants (such as the capitation rate) and variables from last period $t-1$ as the estimates for next period. These values and constants are: The expected frequency of each condition; the effort values per treatment (fixed at the beginning of a simulation); E_{base}; the capitation rate (fixed at the beginning); list size; and fees per treatment (fixed at the beginning). Using these values, he searches for utility maximising values of the choice variables p_i and n_t. The search is implemented using a genetic algorithm. Genetic algorithms are a standard solution for function approximation, and search incrementally for value combinations that come closest

to a fitness value (which is here simply the largest double precision number, as the largest possible GP utility $\rightarrow \infty$).

E_{base} is held constant during this optimisation process. So to speak, it is assumed that the GP sees an illness, which needs to be fixed, and that he has no intrinsic interest in a better doctor-patient relationship. The GP only increases effort if he wants to keep or attract new patients; i.e., he knows that patients value a good patient-doctor relationship, and will use this fact as a 'marketing tool'.

Depending on how many patients are expected per day, the following actions are taken depending on the outcome of the optimisation procedure:

- $n_{opt} < n_{t-1}$: Set $p_{actual} = p_{opt}$ and $n_t = n_{opt}$. In this situation, the waiting list is long enough, and the doctor sees no need to increase the workload. The agent also sets the treatment patterns to the utility-maximising pattern. Furthermore, if $E_{base} > 0$, the GP decreases the base effort by 0.1 (there is no need to attract patients).

- $n_{opt} > n_{t-1}$: The GP wants to have more patients than there is demand. In this case, the agent reacts by re-optimising the optimal treatment patterns p'_{opt} under the constraint that n is given, and sets $p_{actual} = p'_{opt}$. If $E_{base} < 1$, he increases E_{base} by 0.1, because he or she wants to attract more consumers to achieve the preferred workload.

- $n_{opt} = n_{t-1}$: In this case, the situation of the GP remains unchanged. The agent sets $n_t = n_{t-1}$, and all $p_{actual} = p_{opt}$.

To put this model into context with the related literature, it is a model with hidden action - the GP has some discretion whether to treat or refer. There is no diagnosis effort (i.e. no hidden information). The effort variable

is used to represent aspects of the doctor-patient relationship; however, this does not, as in other models (e.g. Jelovac 2001), have a relationship with the correctness of the diagnosis. Furthermore, also in contrast to other models, list size is not a parameter that enters the calculations of the GPs as for example in Grytten and Sorensen (2007). That is, in case of capitation payment, there is no reasoning in the agent that increasing list size by increasing effort can increase income. Rather, this effect would come as a side-effect only if there are too few patients for the doctor's preferred workload.

5.7 Simulations

Simulations are run focusing on different dimensions of the system. The three dimensions considered are: Competition (GP density), payment system, and patient choice. After describing the simulation setup (section 5.7.1.1), first a comparative static perspective is taken by comparing results averaged over all time steps for these dimensions (section 5.7.1.2); then detailed results for a particular GP density are computed and analysed from a dynamic perspective. The dependent variables are waiting lists, referrals, GP effort (as indicator for quality), and patient utility (as indicator for welfare).

5.7.1 Exploration of the Model

5.7.1.1 Parameter Settings and Setup

Pure capitation and pure FFS are the extreme points of prospective and retrospective payment modes. In between these extremes, mixed systems exist. Starting with a pure FFS system, mixed systems are simulated by increasing the capitation rate $I_{capitation}$ from 0 stepwise to 1, at the same

time decreasing the height of fees. So when capitation reaches 1, fees reach 0. Three scenarios are considered: Capitation, half FFS and half capitation, and full FFS.

The effects of (provider-induced) competition are simply represented by simulations with different numbers of GPs.

The effects of patient choice behaviour is realised by running simulations combining small and larger contact networks with rational and probabilistic decision making as described in section 5.6.2: Simulations BR-3 and BR-6 are simulations with a network of 3 and 6 consumers, respectively, using the rational choice mode, i.e. agents collect information, rank it and choose the best expected GP. RL-3 and RL-6 are simulations with a network of 3 and 6 consumers using RL, i.e. a GP is chosen probabilistically. Table 5.1 summarises the model parameters for the simulations. The parameters of the patients' RL action selection function (see equation 2.10) are fixed at $\alpha = 0.1$ and $\gamma = 1$. This is an exploration rate and update speed that enables the patient agents to react in reasonable time to environment changes. The previous chapters illustrate this extensively. α values < 0.1 often lead to suboptimal choices. $\gamma = 1$ leads to the immediate realisation of changes in the environment. Here, this is a change in the doctor's treatment pattern or waiting lists, or the addition of a new GP. If γ is too small, it might take long until the patient realises this change. There is no reason to delay such changes, as there is no noise to accommodate, as, for example, in models where the average reward comes from larger samples of agents.

No variations in GP and patient utility functions are analysed. The only sources of variations are different fees for services, $I_{capitation}$, the geographic distribution, and the learning mode of the patient agents. That is, any resulting differences in the simulation outcome will be based on homogeneous

GP and patient preferences. Any inequalities, say, in the distribution of waiting queues, would be generated simply by the structural and learning properties of the model.

Most of the clinical parameters are set equally in the beginning: There are 3 conditions for which the objective certainty values (p) are fixed. The base fees are drawn from a uniform distribution in the interval 0...1. Then, effort values for each condition-treatment pair $e_{\{m_i,tr_i\}}$ are drawn from a uniform distribution in the range $0...\frac{1}{2}e_{\{m_i,tr_i\}}$ so that effort is always smaller than the fee in the beginning. These base settings are equal for all scenarios. Then, for each payment mode, fees and capitation rate are adjusted. The capitation rate also varies between 0 and 1. For each capitation rate the service fees are decreased by the same amount. For example, if capitation=0.5, then each fee is decreased by 50%, if capitation is 1, then all fees are decreased by 100 %, i.e. set to 0.

Table 5.1 summarises the resulting simulation runs for these parameter settings.

The GP utility function was set in a way that on average GPs prefer a maximum workload per day below the limit of $appointment_{max}$, and are not influenced strongly by patient welfare. Some sample calculations have been made on an Excel sheet to find parameter values for α, β and γ that (on average) first increase the GP's utility until workload becomes so high that utility begins to decrease. In this model, workload is an important endogenous variable responsible for generating variation in the outcomes. Setting the utility function in a way that makes changes in workload unlikely (e.g. by weighting income very high) will induce little variation in the workloads and thus health outcomes. Figure 5.1 illustrates a sample function that was generated with an average fee of 0.3, average effort of 0.5, and an average

Choice	*Pay*	*#pat.*	*#GP*	*cap.*	fee_{avg}	p_{avg}	e_{avg}	*decision*	*net.*
BR-3	S-0	3000	10...250	0	0.48	0.53	0.24	rational	3
	S-0.5	3000	10...250	0.5	0.24	0.53	0.24	rational	3
	S-1	3000	10...250	1	0	0.53	0.24	rational	3
BR-6	S-0	3000	10...250	0	0.48	0.53	0.24	rational	6
	S-0.5	3000	10...250	0.5	0.24	0.53	0.24	rational	6
	S-1	3000	10...250	1	0	0.53	0.24	rational	6
RL-3	S-0	3000	10...250	0	0.48	0.53	0.24	prob.	3
	S-0.5	3000	10...250	0.5	0.24	0.53	0.24	prob.	3
	S-1	3000	10...250	1	0	0.53	0.24	prob.	3
RL-6	S-0	3000	10...250	0	0.48	0.53	0.24	prob.	6
	S-0.5	3000	10...250	0.5	0.24	0.53	0.24	prob.	6
	S-1	3000	10...250	1	0	0.53	0.24	prob.	6

Table 5.1: Overview of simulation runs. The first two columns denote scenario names used in the analysis.

treatment probability of 0.5.

Patients have a simpler objective function: They value the doctor's effort highest and distance the least, whereas the weight of waiting time lies between the two. The function is linear: The higher the effort, the smaller waiting time and distance, the higher utility.

Table 5.2 shows the respective parameters of the utility functions.

Parameter	GP	Patient
	$U_{gp} = I^{\alpha}(E_{max} - E)^{\beta}(n - DEV)^{\gamma}$	$U_P = \alpha wait + \beta dist - \gamma E$
α	0.2	0.5
β	0.8	0.3
γ	0.1	0.7

Table 5.2: GP and patient utility functions

Consumers become ill with probability 0.9. This is certainly unrealistic, but within this simple model justifiable, because time is only relevant for patients to collect experiences about doctors. The value is < 1 to keep some

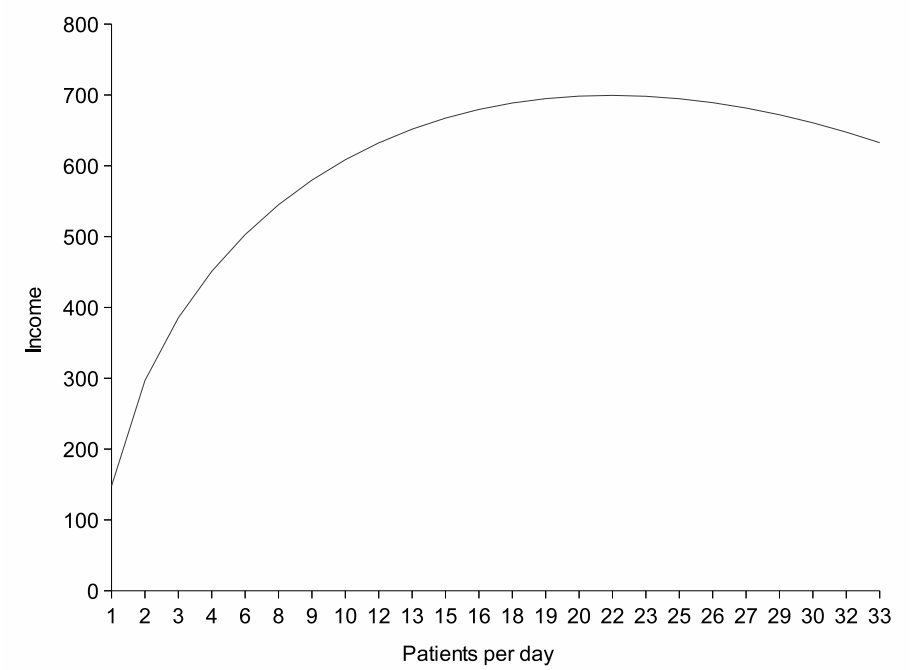


Figure 5.1: A typical GP utility function (showing only the two dimensions income and workload) as used in the simulations.

small variation in the number of ill consumers each time step. Thus, more realistic lower morbidity rates and longer simulation runs are equivalent to higher morbidity and shorter runs. A similar technical reason applies for the decision cycles of GPs, which is set at a week, i.e. $t = 7$ (whereas a period of a month or several months is much more realistic - see, for example, the Krasnik study (Krasnik and Groenewegen 1992), which found more significant changes only after 12 months after the intervention) - the only function is to make the simulation runs more efficient by bundling the important events.

For the comparative static view, simulations were run with 3000 consumers and varying GP density. The reason for this large number was mainly to be absolutely certain that the geographical distribution of consumers is random. Only 120 time steps were run. The main interest is

exploring many different competition scenarios by increasing GP density and generating large enough samples of distribution parameters such as fee and effort values. To keep the time to compute the simulations manageable, the duration of the simulations was kept short - therefore the high morbidity rate and shorter decision periods described above. For the dynamic view, the number of patients was reduced to 1000 and only one GP density setting run, which, comparing it with the larger simulation, seemed to be a sufficient size. On the other hand, the number of time steps was increased to 750 to observe the behaviour of the model in the longer run.

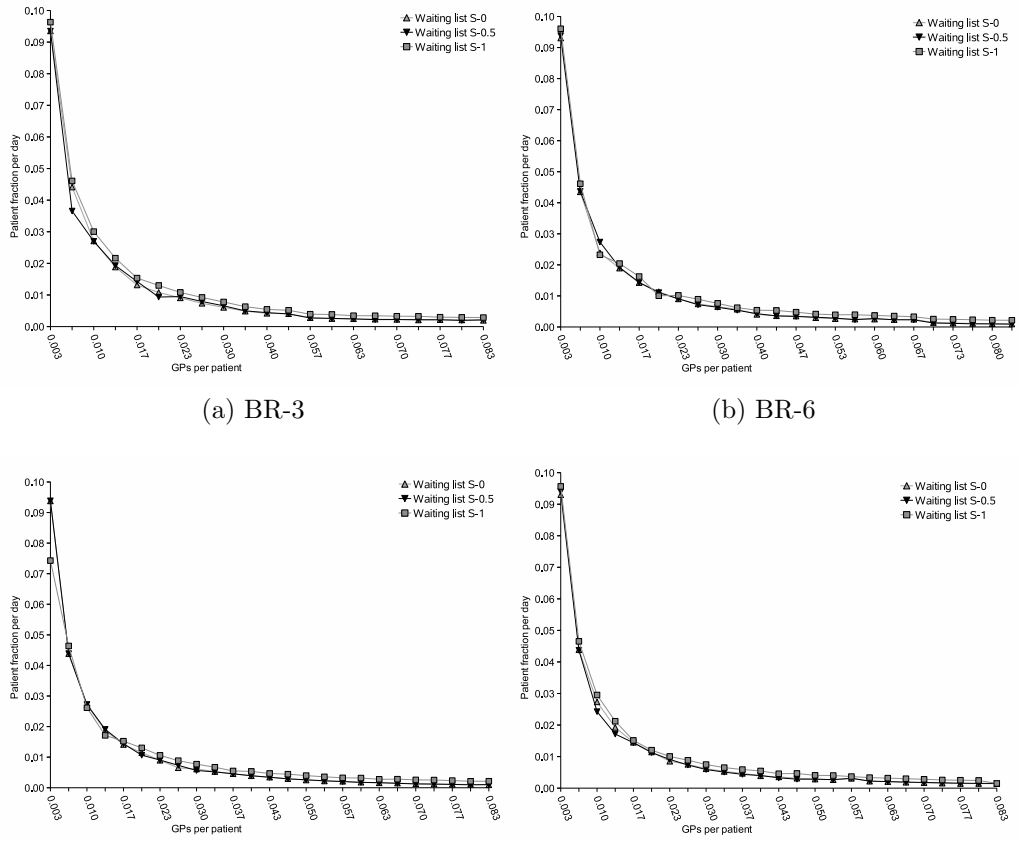
5.7.1.2 Static Analysis

The following sections show simulation runs averaged over 121 time steps, representing 120 days or four months. In the figures, GP density is measured as the quotient of the number of GPs and patients in the respective simulation.

Waiting lists

Figure 5.2 illustrates how GP density, payment system and choice mode influence waiting time. Waiting lists are measured as the quotient of patients waiting for treatment at a time step, and all consumers in the population. Quite trivially, waiting lists decrease with increasing GP density in all scenarios.

Comparing the scenarios, there is a very small difference between pure capitation systems and non-capitation systems; there is virtually no difference between a pure FFS system and the mixed half-fee, half-capitation system. Furthermore, there seem to be some very small differences between choice modes. In the BR-6 scenario, for example, a GP density of 0.01



(a) BR-3 (b) BR-6

(c) RL-3 (d) RL-6

Figure 5.2: Waiting lists (static analysis)

induces average waiting lists of 0.025. This value is smaller than the BR-3 scenario (where less consumer choice is possible), as well as all RL scenarios.

Referrals

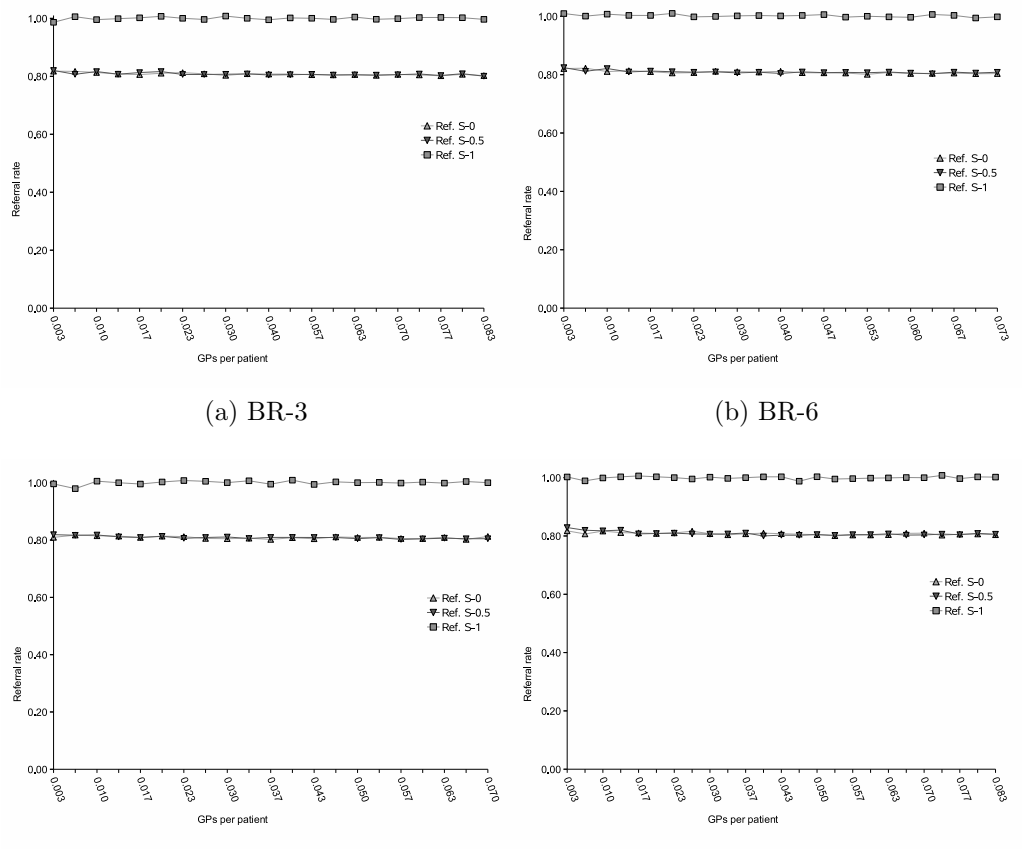
Figure 5.3 shows the referral behaviour. The rate of referrals r is computed relative against the expected referrals, which is determined by the certainty value p: $r = \frac{1}{p}\frac{\#referrals}{\#referrals+\#treatments}$. For example, in the simulation setup the 'objective' treatment probability is $p = 0.53$ and referral probability $= 1 - p = 0.47$. If the doctors' decisions are on average close to 0.47, then $r \approx 1$ and vice versa. The maximum deviation parameter var_{max} was set

to 0.2, so that r varies between 1.2 and 0.8.

There is a clear difference between capitation and non-capitation systems, but not between choice modes. r is almost constant for capitation doctors at a rate close to 1; they decide 'close to the norm'. This rate does not change with increasing competition. This is somewhat surprising as the major instrument for capitation doctors to increase utility is the reduction of effort ($\beta = 0.8$). Patient welfare should not have such a big influence ($\gamma = 0.1$); one would, therefore, expect a propensity to refer closer to the maximum of 1.2 in all scenarios. In the FFS scenarios, the referral rate is constant at about 0.8 - close to the minimum possible. The incentives of the model are such that GPs always decide to over-treat; the rationale of an FFS agent is to maximise income at each consultation independent of the environment. This is plausible, as there is no incentive apart from patient welfare to increase the referral rate. While patient welfare can influence the decision of capitation doctors, resulting in appropriate treatment, this influence is (ceteris paribus) too weak for FFS GPs in the model.

Effort

As could be expected, effort levels (figure 5.4) increase with GP density as increasing effort is the main instrument for doctors to attract more patients. Shape and level are similar in all scenarios, although effort is ends up slightly higher in the BR models (≈ 0.9 as compared to ≈ 0.8 in RL). Furthermore, in the BR-6 scenario under capitation is only very small change in effort; it starts relatively high and then remains similar.



(a) BR-3

(b) BR-6

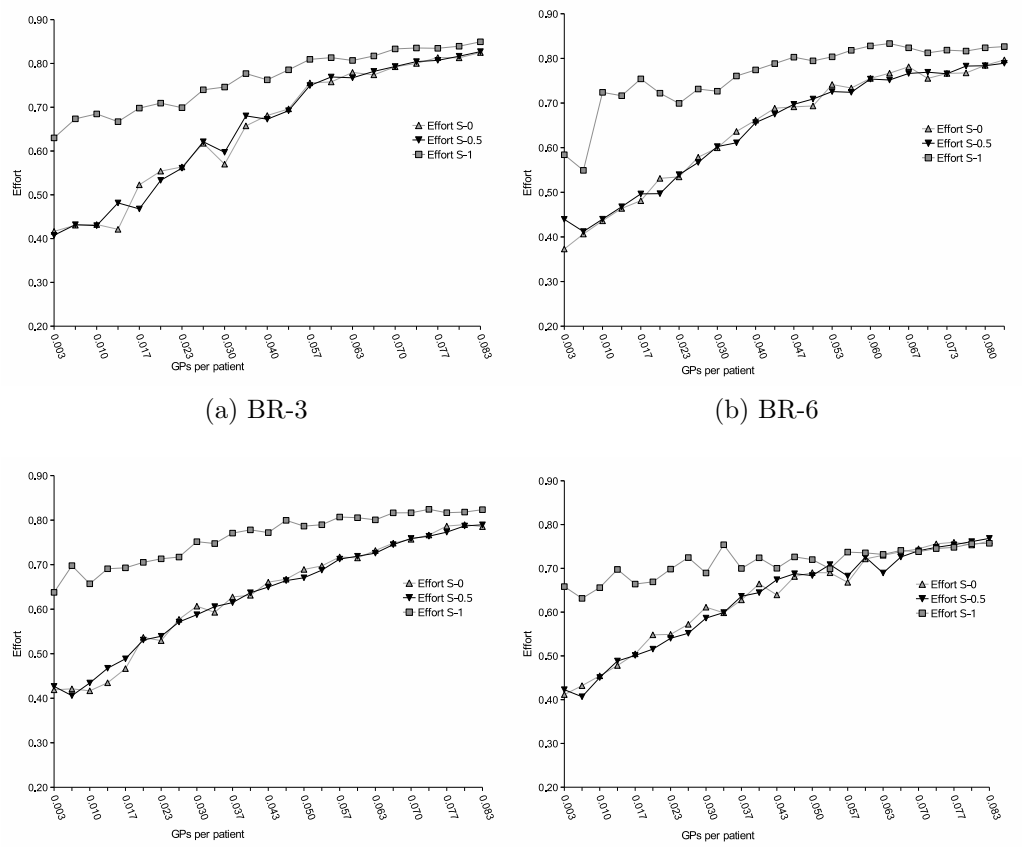
(c) RL-3

(d) RL-6

Figure 5.3: Referrals (static analysis)

Patient utility

More differences between the scenarios exist with respect to patient utility (figure 5.5). In BR-3, patient utility increases at a decreasing rate up to a level of about 0.6. It then remains, by and large, at this level and seems even to decrease when GP density increases further over ≈ 0.063. In BR-6, the rise in utility is more constant as competition increases, and at the top with 0.7 higher than in BR-3. Utility in the RL scenarios is lower, but increasing almost linearly with GP density.



(a) BR-3 (b) BR-6

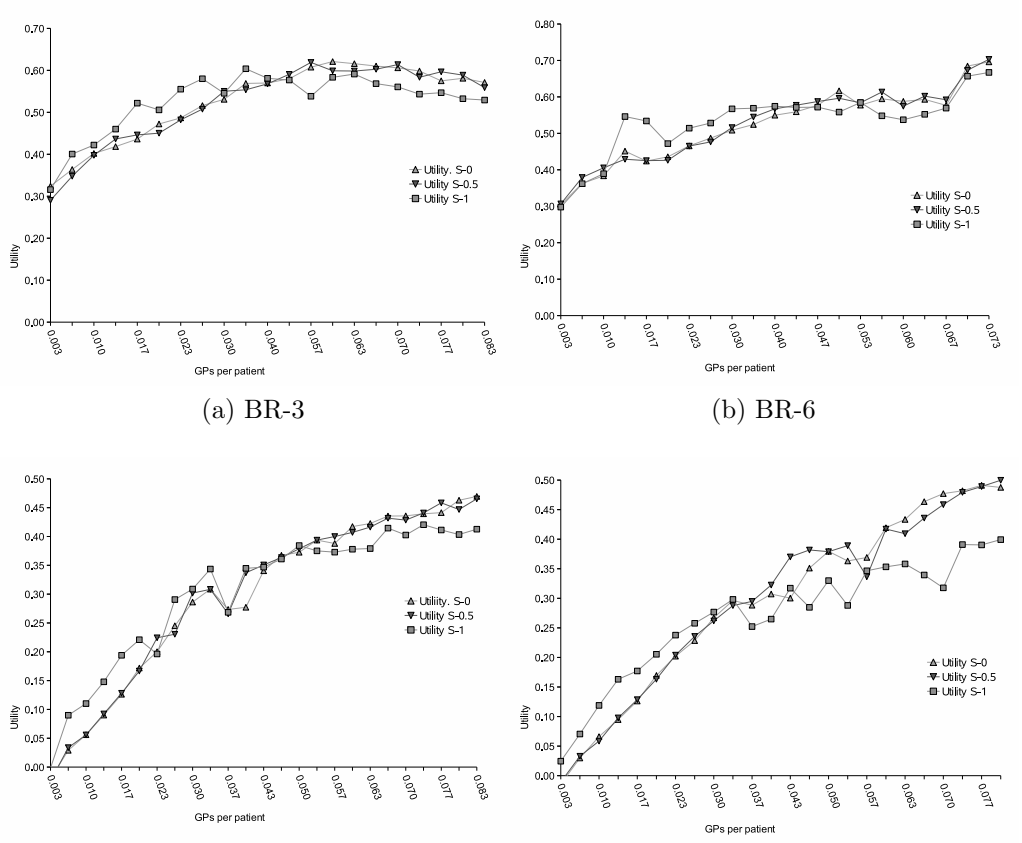
(c) RL-3 (d) RL-6

Figure 5.4: GP effort (static analysis)

Summary

This overview showed on a coarse level the simulation outcomes in the dimensions GP density and patient choice behaviour. There are mostly no or only very little differences between the behaviour modes. Furthermore, the relationship between competition, effort and welfare is obvious - the more competition the higher quality. The following two main observations will be investigated in more detail in the next set of experiments:

– Less rational choice behaviour (in the form of RL) leads to reduced effort.



(a) BR-3 (b) BR-6

(c) RL-3 (d) RL-6

Figure 5.5: Patient utility (static analysis)

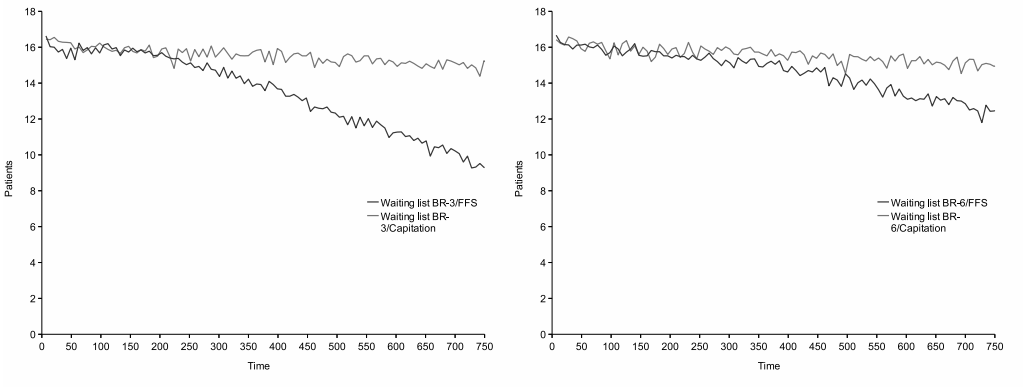
– Patient welfare increases with better information (larger networks) and more rational decision-making. There are two possibilities why this can happen - either BR patients switch faster to better doctors even if differences are very small; or, vice versa, the probabilistic choices under RL lead to increased switching (probability matching) and apparent random behaviour. In the latter case, this would induce doctors to reduce effort, since it does not necessarily increase list size; hence reducing effort could be a suitable strategy to improve GP utility.

5.7.2 Dynamic Analysis

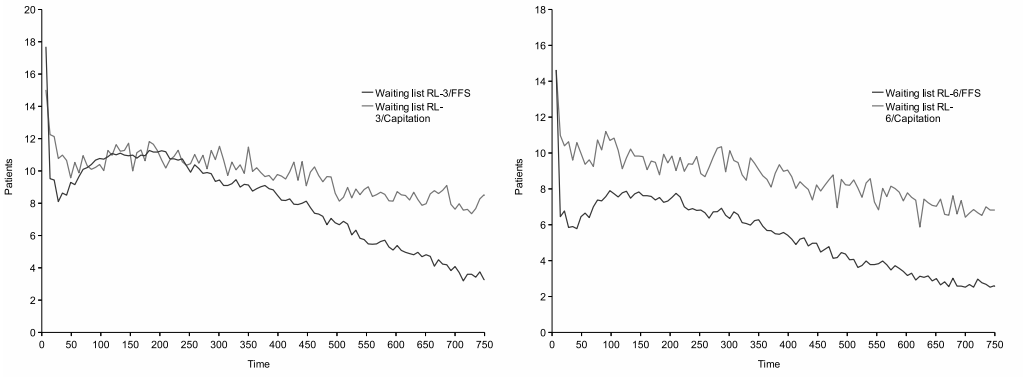
The purpose of the preceding section was to cover many different parameter variations on an aggregate level. While the aggregate view only gives a general impression of the simulation behaviour, this section takes the exploratory results as a starting point and looks at the dynamic aspects. For this, GP density is fixed and the number of patients reduced to obtain results in a reasonable time. The number of patients is set at 1000, the number of GPs at 60. Simulations are run for 750 steps. Furthermore, only pure FFS versus pure capitation is compared, since the effects of minor variations turned out to be negligible in the model. The other parameters remain the same. The GP-patient ratio reflects the actual ratio in the UK. According to a 2003 OECD report (cited in Royal College of General Practitioners (2006)) this ratio was 65 GPs per 100.000 patients, of whom the average patient had three visits per year. Given the high probability of visiting a doctor in the simulation setup (0.9 as compared to 0.008), this ratio of 0.000065 can be translated into a ratio of 0.06 in the simulation, with the average workload of the GP remaining about the same.

Waiting lists

The development of waiting lists (figure 5.6) shows an important aspect that was not observable during the shorter runs of the previous simulations: Waiting lists decrease only at later stages of the simulation - roughly from step 250 onwards. Over all learning modes, the decrease is sharper for FFS; for network sizes of 6, the difference is smaller. Furthermore, the difference between BR and RL modes is large. Waiting lists drop much faster to low levels under RL. In RL-6 waiting lists are generally shorter than in RL-3.



(a) BR-3 (b) BR-6



(c) RL-3 (d) RL-6

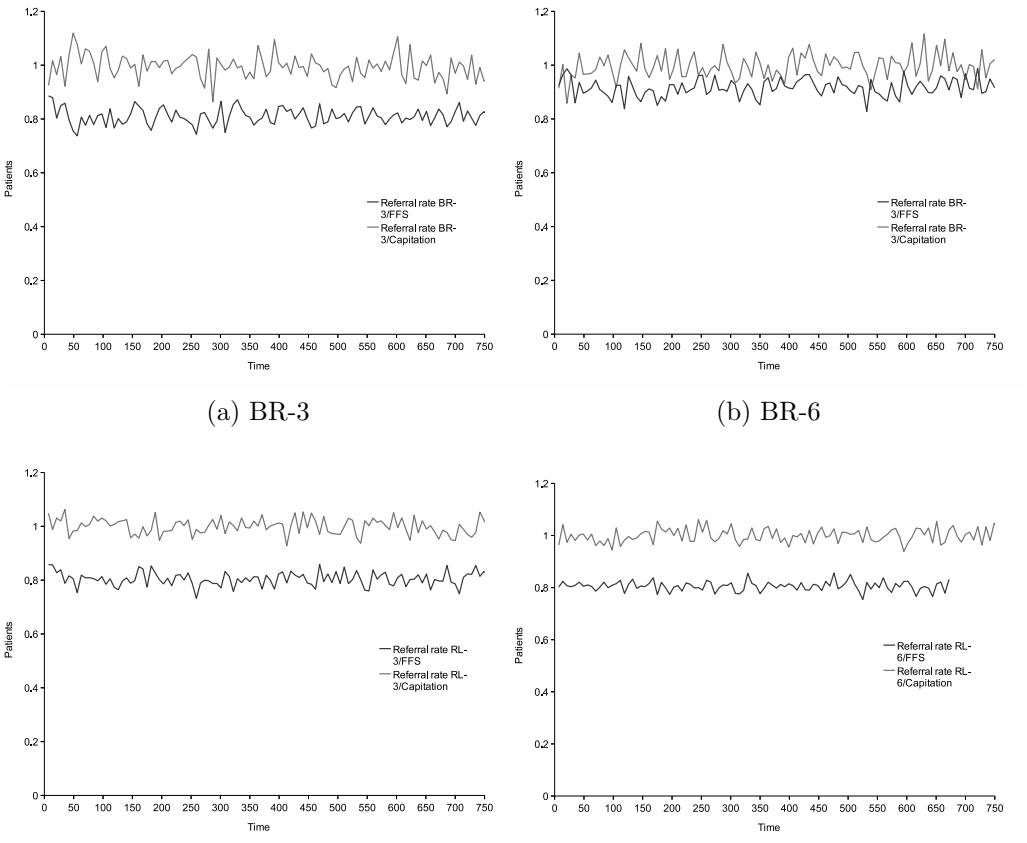
Figure 5.6: Waiting lists over time for a density of 0.06 GPs per patient.

Referrals

Figure 5.7 shows the referral behaviour. Here, the differences are very small and the rates stable. However, differences between FFS and capitation in BR-6 are smaller. Furthermore, variation seems to be larger in BR scenarios.

Effort

Figure 5.8 shows the effort levels. Effort increases quickly to the maximum of 1 in all scenarios. The rise is a little slower in the RL scenarios. Furthermore, there is - although very little - variability in effort levels under capitation in the RL-scenarios, and the level is - also only slightly - below



(a) BR-3 (b) BR-6

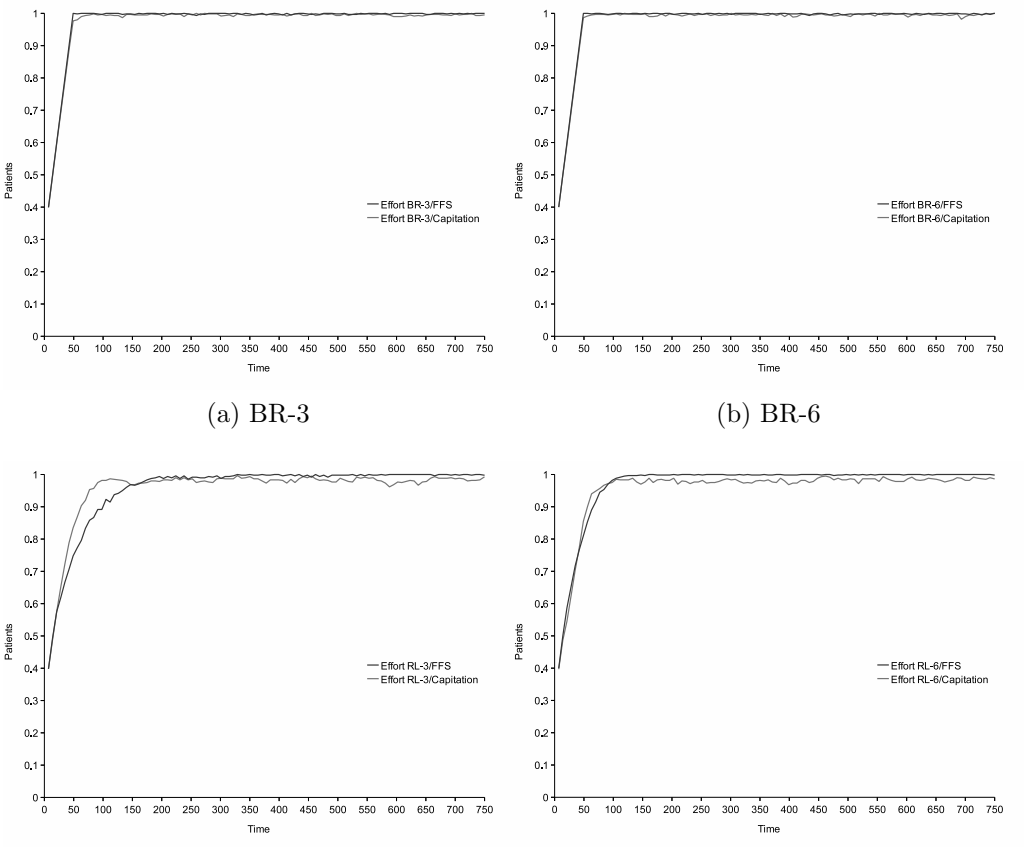
(c) RL-3 (d) RL-6

Figure 5.7: Referrals over time for a density of 0.06 GPs per patient.

the FFS levels.

Patient utility

Figure 5.9 shows the patient utility levels. Here, there are very obvious differences between choice modes. The BR scenarios are very similar - neither network size, nor payment mode influences patient utility strongly. The difference is mainly that utility reaches its maximum a few time steps earlier in the BR-6 scenarios, and this even faster under FFS. Within the RL scenarios, patient utility is lower and varies much stronger. Moreover, in the RL scenarios, utility is lower and variation stronger under capitation



(a) BR-3 (b) BR-6

(c) RL-3 (d) RL-6

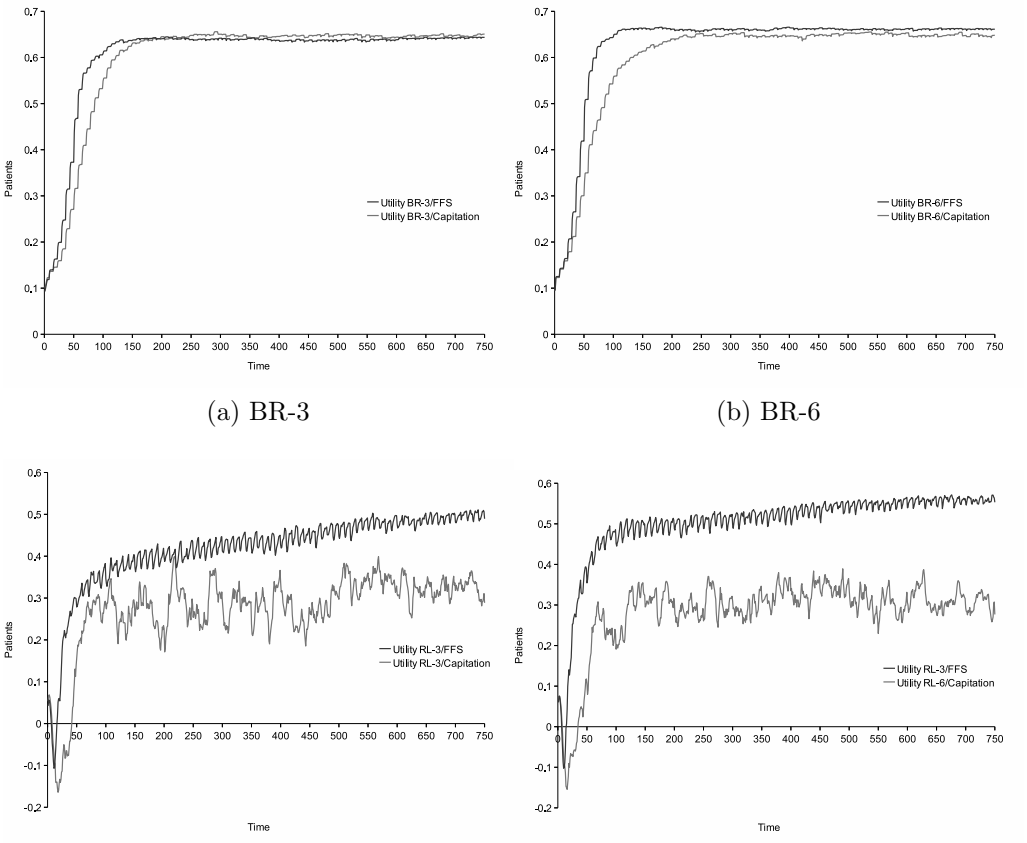
Figure 5.8: GP effort over time for a density of 0.06 GPs per patient.

than under FFS.

5.7.3 Summary of the Simulation Results and Discussion

The dynamic view highlights the driving factors in the simulation in a more detailed view. With regard to waiting lists, it was found that there are considerable differences between choice modes and payment system. With respect to patient welfare, there are differences between choice modes.

Table 5.3 shows some summary measures across all time steps, high-



(a) BR-3 (b) BR-6

(c) RL-3 (d) RL-6

Figure 5.9: Patient utility for a density of 0.06 GPs per patient.

lighting some differences between learning modes and payment systems. A variance analysis for the dependent variables waiting list, effort, referral rate and patient welfare has been conducted; results are given in the appendix C.

Using the graphs and the results of the variance analysis, two main observations can be made:

In the simulations, patient choice reduces waiting times, but decreases quality: For waiting lists, the variance analysis shows significant differences between payment systems and choice mode (except for BR-3/Capitation

scenario	*effort*		*wait*		*referral rate*		*patient utility*	
	mean	*sdev*	*mean*	*sdev*	*mean*	*sdev*	*mean*	*sdev*
BR-3/FFS	0.982	0.000	9.625	5.711	0.808	0.017	0.617	0.053
BR-3/Capitation	0.976	0.005	13.944	5.518	0.988	0.035	0.611	0.053
BR-6/FFS	0.980	0.003	11.636	4.113	0.911	0.023	0.637	0.027
BR-6/Capitation	0.976	0.003	14.070	6.037	0.991	0.029	0.612	0.04
RL-3/FFS	0.963	0.016	6.651	2.497	0.803	0.013	0.44	0.123
RL-3/Capitation	0.964	0.008	7.709	2.229	0.993	0.020	0.289	0.157
RL-6/FFS	0.960	0.010	5.145	2.304	0.805	0.020	0.454	0.155
RL-6/Capitation	0.956	0.015	8.666	3.330	0.998	0.020	0.287	0.189

Table 5.3: Mean and standard deviation for dependent variables, measured over 750 time steps.

which has no significant differences to RL-3/Capitation and RL-6/Capitation; this is probably due to the strong decrease at later simulation steps). For patient utility, the differences between choice mode and payment system are also significant, with two exceptions (BR-3/Capitation is not different from BR-3/FFS and BR-6/Capitation).

A likely reason for these clear differences between choice modes lies in GP effort and patient mobility. In BR scenarios, patients remain loyal to their GPs. This results in high and unchanged effort levels. Since effort is strongly weighted in the patient utility function, this would explain the lower utility levels in the RL models. Differences in effort levels are small, but significant between BR and RL scenarios. Furthermore, significant differences in effort exist between RL-6/Capitation and all other RL scenarios. With respect to waiting lists, computations showed that the coefficient of variation in the RL-scenarios is much lower (on average 0.37) than in BR-scenarios (on average 0.44). This indicates that in BR there are some doctors with longer, and some with short waiting lists, while in RL, patients distribute more evenly over practices.

Strictly choosing the most preferred doctor thus guarantee stability in demand for the GPs, who in turn have a motivation to maintain quality to keep patients. Both demand and supply stabilise each other. As the differences between GPs are small (due to homogeneous utility functions fixed in the setup), RL patients tend to go 'GP shopping'. The closer expected utility the more likely they switch. This trend is also obvious when computing a loyalty index (table 5.4) as the ratio of the number of visits at the GP most often seen by a patient and the number of total GP visits. It shows that patients are most loyal to their GP in FFS scenarios with rational decisions and small network size. In other words, limited choice and economic decisions stabilise the system best. For example, in BR-3/FFS 74% of all patients stayed with a single GP, whereas in all RL scenarios, this rate is just 30%. The difference between capitation and FFS in the BR models is likely due to shorter waiting lists: because capitation doctors have longer waiting lists, some patients become unsatisfied faster than under FFS and may switch. This would also be reflected by the smaller differences between waiting lists in BR-6 as compared to BR-3, as are the differences between loyalty values. Another pointer into this direction can be seen in the referral behaviour. In BR-6, for example, even FFS doctors tend to refer at more appropriate levels (which increases patient utility), rather than over-treating.

While patient utility increases due to higher quality in BR scenarios, longer waiting queues develop at 'good' doctors, while others have only few patients. This shows the tension between policy goals - less choice might actually provide increase quality and welfare, but waiting times are likely to increase. More patient choice could reduce waiting lists, but also the quality of doctor-patient relationship.

In the simulations, FFS appears to reduce waiting times better than cap-

Scenario	*Loyalty*
BR-3/FFS	0.75
BR-3/Capitation	0.49
BR-6/FFS	0.53
BR-6/Capitation	0.48
RL-3/FFS	0.28
RL-3/Capitation	0.29
RL-6/FFS	0.29
RL-6/Capitation	0.29

Table 5.4: Average loyalty index of patients. The index is computed as the quotient of the number of visits at a most visited GP and total visits of GPs over the whole simulation.

itation: Looking only at the influence of payment systems, the simulations suggest that the FFS scenarios score better on most dependent variables than capitation scenarios.

With respect to waiting lists, the reason is that the preferred workload of FFS doctors is always higher (computations show that the preferred workload of FFS doctors is roughly twice the preferred workload of capitation doctors). As the preceding figures showed, they increase their utility by increasing income by treating more patients, and referring fewer patients. Consequently, they have on average shorter waiting lists than capitation doctors. This effect is not visible in early stages of the simulation; only over time doctors manage to decrease their waiting lists by adapting their planned workload week after week as their queues increase. It also explains the different speed of BR-3 and BR-6: In BR-6 patients know more doctors (coefficient of variation for waiting lists: 0.43) than in BR-3 (coefficient of variation for waiting lists: 0.4), and thus distribute more evenly over the GP population. The same tendency, but on different levels, is obvious from figure 5.6 for the RL scenarios.

Looking at patient welfare, utility is always lower in the capitation cases - slightly in the BR scenarios, and more obvious in the RL scenarios. In the BR scenarios, this difference can easily be attributed to the longer waiting lists, which have some influence on utility. The large differences in RL are, however, puzzling. Certainly also here the longer waiting lists influence welfare negatively. The question is why level and shape of the curves differ so much and persistently. The pattern in the first steps is analogous to the BR simulations - under capitation welfare is always slightly lower. However, over time, the difference between utility stabilises, instead of approximating each other as under BR. The only remaining source of variation remains GP effort. Effort does vary more and is lower as under FFS, but the difference is extremely small (and not significant within RL-3).

Similar to the first result, this implies that FFS might be more efficient in reducing waiting times. Moreover, if there is high patient mobility, more FFS counterbalances the welfare loss due to lower effort levels.

5.8 Conclusion

Based on the results of the literature about incentive systems in primary care, an agent-based model was developed, which attempted to address the major shortcomings of the traditional models: The simplifying assumptions of the agency literature where GPs are modelled as income-maximising firms, and the ad-hoc nature and assumptions of many empirical studies. In particular, the model tried to operationalise patient choice, which plays an important role in the political discussion, but about which few models exist.

The simulations demonstrated how the impacts of possibly conflicting policy targets (quality and efficiency) can be analysed within one and the

same model. The model shows that more 'shopping for GPs' could, while reducing waiting times, actually lead to lower quality and consequently, to lower patient utility. The underlying reason is that GPs might choose to work less in an environment they perceive as unpredictable and unstable. If policy values low waiting times higher, these negative effects might be accepted, because more shopping is likely to lead to a more equal distribution of patients over GPs. The simulations also showed how the interactions between payment system and patient behaviour might be analysed. In particular, in the light of these artificial results, it could be argued, that - if efficiency and quality are equally weighted goals - the implementation of more patient choice should be accompanied with more FFS-like elements.

The result of the simulations could also be interpreted in a different way. Assuming that policy 'wants' educated consumers behaving as rational as the consumers in the BR-scenarios, the scenarios can be used as a thought experiment of possible future paths. Consumers, prepared to make the best choice, search for their preferred practice. They want to behave rationally in the sense defined above, i.e. stick to the best GP and not shop around. However, differences between practices, e.g. in a certain region, are so small or there is not enough reliable information, that it is too difficult to distinguish between doctors' quality. It becomes impossible to find the best GP (this is represented by the RL scenarios). As patients keep searching for better practices, quality levels fall because GPs see no reason why they should raise effort for non-loyal patients. Thus, even if policy could reach the objectives in one area - motivating consumers to exercise choice as a means to raise quality - the actual achievement of this goal might lead to unintended consequences.

Chapter 6

Conclusion

This thesis focused on adaptation in artificial agents from different angles: First, how simple and more complex approaches to learning and cognition can be combined in an ACE framework; how simple and more complex learning can be applied in various domains; and how a software can be engineered that covers the implementation of this rather diverse set of issues.

Summary of results Chapter 2 showed that it is possible to learn about the environment an agent lives in with very little a priori knowledge. The main idea of the approach presented was an incremental search for the best state-action mappings in the state space. Behaviour is learnt in a trial- and error fashion using reinforcement learning, and then by mapping the action selection probabilities to state descriptions. These mappings are similar to simple rules. Which descriptions are generated depends on their relevance, or what the agent 'decides' to know about its environment. This way, agents learn to distinguish between important and less important details of the world they live in. A simple experiment illustrated how the algorithm works. The BRA algorithm is different from many learning approaches in ACE as it combines rule learning and reinforcement learning in a dynamic

way. BRA is a general approach, and it can cover a variety of approaches in the simulation literature. For example, it is very similar to learning classifier systems, but also able to represent very simple forms of learning as well.

Chapter 3 applied the algorithm to a model of statistical discrimination. The aim was to build a bridge between theoretical game theory and the classroom game conducted by Fryer Jr. et al (2005). It was shown that the RL model is capable of reproducing the empirical results as well as the behavioural patterns observed in the experiment. A further parallel is that in both simulations and experiment discrimination is rare. In the simulations, no general rule or scenario was found that generated discrimination on the average. In most cases, discrimination either did not evolve, or disappeared in the longer run. In the samples where discrimination was observed, the occurrence seemed path-dependent. In particular, if employers are liberal in the beginning and differences in worker productivity are persistent, employers adjust their hiring levels eventually and a discriminatory outcome emerged.

Chapter 4 developed a RL specification of a communication network model. In the base version, it became clear that RL produces similar results as theoretical predictions. It was furthermore shown that the simplest possible RL model is sufficient to produce that result. Using BRA, no plausible mapping from player names to actions was found that was superior than simple stimulus-response learning. Aggregate results did also not improve. Comparing the RL model with an experiment from behavioural game theory, it turned out that the model predicts the empirical results better than the equilibrium prediction. This result has analogies to earlier research in behavioural game theory, which often finds that the simplest model fit actual data and theoretical results reasonably well; sometimes even better than more complicated models. What is different here is that the RL con-

nections model can state this also for the more complex class of network games.

In chapter 5, agents were much simpler. The purpose of the health care simulations was to use agent-based modelling to investigate models of a complex environment. The primary care sector can be seen as such complex environment. Some authors already made a case for applying complexity science tools in this area. The simulations looked at some hypotheses that were formulated and, to a certain extent, tested empirically. Further experiments highlighted the influence of patient information and choice behaviour on health outcomes. The model served two purposes: First, it puts BRA, although in its simplest form, into the context of a more complex model. Second, it explored how ACE could be applied to primary care, for which no other computational approaches exist so far. The results showed that assumptions about patient behaviour influence the simulation result considerably. The main result here is that more consumer choice can lead to worse health outcomes, as doctors have no incentives to provide personalised services to non loyal consumers. Since most debate about the benefits of consumer choice in health care is still driven by ideology, often based on improvable facts about the benefits of competition, an ACE model may be a starting point for a more rigorous analysis of arguments in this area.

Limitations The motivation of this work was to generate aggregate outcomes (sometimes also described with the term 'emergence') like discrimination, health outcomes or network structures by adaptive algorithms. The nature of complex systems and some definitions were introduced in the introduction (see definitions 1 to 3). Of the three presented definitions, the aspects covered in this work matches only the first two. The agents were goal directed (utility optimisers) and reacted to changes in their environ-

ment; they are not active planners in order to achieve some (sub-)goals, which would require at least some representation of plans and goals as well as capabilities to reason about them. For the type of models discussed in this thesis, this is not necessary. More precisely, the overall approach taken here is based on simple types of learning. The basis of this approach has been tied to existing concepts of bounded rationality. However, there are cases where more complex models of cognition and goal-directed behaviour are necessary. For instance, as Gilbert (2006) mentions, agents in team environments may need to hold cognitive models about their colleagues and develop strategies to improve the performance of the team as a whole. Although BRA provides a simple cognitive representation in the form of rules and symbolic state descriptors, it cannot handle such cases. For example, a network game with farsighted players as, e.g., in Watts (2002) or Deroian (2003) is already difficult to represent with BRA, as agents would need an idea of what networks might form, how other agents are likely to act and so on.

The BRA approach is thus useful for classification problems or where cases can be translated into such classifications - hence it has a close relationship to classifier systems. The discrimination game in chapter 3 is a representative application - employers have to classify two types of worker agents and behave accordingly. In other domains, classification may simply be not necessary. For example, the primary care model in chapter 5 can formally be modelled with the framework, but since there are no classification problems to solve, the mechanism reduces to simple RL. From a different angle, chapter 4 showed that where classification (based on the labels of players) is a model option, it might simply not add anything to the quality of the model result.

Another limitation of this thesis is its relationship to empirical valida-

tion. The models of chapters 3 and 4 were only loosely coupled to experiments from behavioural game theory. The main interest was to develop models with learning agents and to use BRA framework for this. The primary care model in its current form is too general to be fitted to existing data. More work to assemble the necessary data and to fit the model structure to it is required first.

Future work Future work should therefore focus on two aspects: The empirical validation of the implemented models, and the development of richer applied models to make better use of the BRA features. Looking back at the main criticisms presented in the introduction, these are probably two conflicting goals. The richer the model, the more likely the results produced with it are less general, and that it is only one of many models with which the empirical fact can be explained.

Looking at validation, the following paths are possible: In the area of network games, more models for similar games with endogenous network structures can be devised where experimental data is already available; the comparison with Conte et al (2009) is an initial step into this direction. Model parameters could be calibrated in a way that produces a minimal deviation from the actual, empirical outcome. This is certainly more difficult in a health care model that is inherently related to real-world processes. Here, a main path will be the collection of appropriate data, e.g. on regional levels, using this data first for the specification of input parameters (geographical distribution, preferences, consumer types, etc.), and only then for comparing artificial with real results (e.g., comparison of mobility rates, GP lists, waiting lists, etc.). To what extent this procedure is possible depends on the availability of data. Another aspect of calibrating the model to actual health systems is to map the various levels of real health systems

better and to model the consequences that arise from there. The model restricted health provision to individual doctors, however, different organisational forms exist. For example, GPs are organised on the practice level; Primary Care Trusts (PCT) organise practices and so on. Furthermore, the model deliberately modelled GPs as self-employed; nevertheless, GPs could act also (partially) as firms. So, under the PCT scheme, GPs can invest their surpluses into their practices. This adds another dimension to the utility function not covered in the model.

As empirical validation can add to the quality of the models discussed in this thesis, so could more complex models make use of the features of BRA. For instance, rule learning could be added to the primary care model: Patients can learn to distinguish dynamically between doctors for different illnesses and build rules which doctors or specialists to consider under different conditions. Conversely, doctors might learn rules in which area they want to specialise, depending on the demand for certain health services. With this, a model of provider specialisation could be built.

Bibliography

Altonji JG, Blank RM (1999) Race and Gender in the Labor Market. In: Ashenfelter O, Card D (eds) Handbook of Labor Economics, volume 3C, Amsterdam,Oxford: Elsevier, chap 48, pp 3143–3260

Anderson DM, Haupert MJ (1999) Employment and Statistical Discrimination: A Hands-On Experiment. The Journal of Economics 25(1):85–102

Anderson J (1993) Rules of the mind. Lawrence Erlbaum Associates Inc., Hillsdale

Anderson LR, Fryer RG, Holt CA (2005) Discrimination: Experimental Evidence from Psychology and Economics. In: Rogers W (ed) Handbook on Economics of Discrimination

Appleby J, Dixon J (2004) Patient choice in the NHS - Having choice may not improve health outcomes. British Medical Journal 329(7457):61–62

Arrow (1963) Uncertainty and the Welfare Economics of Medical Care. The American Economic Review 53(5):941–973

Arrow K (1973) The theory of discrimination. In: Ashenfelter O, Rees A (eds) Discrimination in Labor markets, Princeton University Press, Princeton, NJ

Arthur WB (1993) On designing economic agents that behave like human agents. Journal of Evolutionary Economics 3(1):1–22

Ascape (2010) The Ascape project, URL http://www.brook.edu/es/dynamics/models/ascape/, last accessed June 2010

Auman R (1997) Rationality and Bounded Rationality. Games and Economic Behaviour 21:2–14

Bagnall AJ, Smith GD (2005) A Multi-Agent Model of the UK Market in Electricity Generation. IEEE Transactions on Evolutionary Computation 9(5):522–536

Bala V, Goyal S (2000) A Noncooperative Model of Network Formation. Econometrica 68(5):1181–1229

Barabasi AL, Albert R (1999) Emergence of Scaling in Random Networks. Science 286(5439):509–512

Beal S, Querou N (2007) Bounded rationality and repeated network formation. Mathematical Social Sciences 54:71–89

Becker GS (1957) The Economics of Discrimination. The University of Chicago Press, Chicago

Beggs AW (2005) On the convergence of reinforcement learning. Journal of Economic Theory 122:1–36

Bendor J, Mookherjee D, Ray D (2001a) Aspiration-based Reinforcement Learning in Repeated Interaction Games: An Overview. International Game Theory Review 3(2-3):159–174

Bendor J, Mookherjee D, Ray D (2001b) Reinforcement Learning in Repeated Interaction Games. Advances in Theoretical Economics 1(1):1–42

Bernasconi M, Galizzi MM (2005) Coordination in Network Formation: Experimental Evidence on Learning and Salience, FEEM Working Paper no. 107

Blume LE (2006) The Dynamics of Statistical Discrimination. The Economic Journal 116:F480–F498

Boergers T, Sarin R (1997) Learning Through Reinforcement and Replicator Dynamics. Journal of Economic Theory 77:1–14

Boergers T, Sarin R (2000) Naive learning with endogenous aspirations. International Economic Review 41(4):921–950

Bondy JA (2008) Graph theory. Springer, London

Brenner T (2006) Agent learning representation: Advice on modelling economic learning. In: Tesfatsion L, Judd K (eds) Handbook of Computational Economics, 2, Elsevier, chap 18, pp 896–942

Bush R, Mosteller F (1955) Stochastic Models for Learning. Wiley, New York

Butz M (2002) An algorithmic description of ACS2. In: Lanzi P, Stolzmann W, Wilson S (eds) Advances in learning classifier systems, Lecture Notes in Artificial Intelligence, Springer, Berlin, vol 2321, pp 211–229

Callander S, Plott C (2005) Principles of Network Development and Evolution: An Experimental study. Journal of Public Economics 89(8):1469–1495

Camerer CF, Ho TH (1999) Experience-weighted attraction learning in normal form games. Econometrica 67(4):827–874

Camerer CF, Chong JK, Ho TH (2007) Self-tuning experience weighted attraction learning in games. Journal of Economic Theory 133:177–198

Cecconi F, Parisi D (1998) Inividual versus social survival strategies. Jornal of Artificial Societes and Social Simulation 1(2)

Chalkley M, Malcomson J (1998a) Contracting for health services when patient demand does not reflect quality. Journal of Health Economics 17:1–19

Chalkley M, Malcomson J (1998b) Contracting for health services with unmonitored quality. Economic Journal 108(449):1093–1110

Chen Y, Khoroshilov Y (2003) Learning under limited information. Games and Economic Behavior 44:1–25

Chen Y, Tang FF (1998) Learning and Incentive-Compatible Mechanisms for Public Goods Provision: An Experimental Study. Journal of Political Economy 106(3):633–662

Cicirelli F, Furfaro A, Giordano A, Nigro L (2009) Distributed Simulation of RePast Models over HLA/Actors. In: Turner SJ, Roberts D, Cai W, El-Saddik A (eds) DS-RT, IEEE Computer Society, pp 184–191

Cioffi-Revilla C, Luke S, Panait L, Sullivan K (2004) MASON: A New Multi-Agent Simulation Toolkit. In: Proceedings of the 2004 SwarmFest Workshop

Coate S, Loury GC (1993) Will Affirmative-Action Policies Eliminate Negative Stereotypes? American Economic Review 83(5):1220–1240

Conte A, Di Cagno D, Sciubba E (2009) Strategies in Social Network Formation, Jena Economics Research Papers, Friedrich Schiller University and Max Planck Institute of Economics, Jena, Germany

Davidson S, Manheim L, Werner S, Hohlen M, Yudkowsky B, Fleming G (1992) Prepayment with office-based physicians in publicly funded

programs: results from the Childrens Medicaid Program. Pediatrics 89(4):761–767

Davis DD (1987) Maximal Quality Selection an d Discrimination in Employment. Journal of Economic Behaviour and Organization 8:97–112

Deroian F (2003) Farsighted strategies in the formation of a communication network. Economic Letters 80:343–349

Devlin R, Sisira S (2008) Do physician remuneration schemes matter? The case of Canadian family physicians. Journal of Health Economics 27:1168–1181

Dixon A, Robertson R (2008) Patient choice in general practice: the implications of patient satisfaction surveys. Journal of Health Services Research & Policy 13(2):67–72

Doreian P (2006) Actor network utilities and network evolution. Social Networks 28(2):137–164

Dranove D (1985) Demand Inducement of the Physician-Patient Relationship. Working Paper, University of Chicago

Dutta B, Mutuswami S (1997) Stable Networks. Journal of Economic Theory 76:251–272

Edmonds B, Moss S (2005) From KISS to KIDS? an 'anti-simplistic' modelling approach. In: Davidson P (ed) Lecture Notes in Artificial Intelligence: Multi Agent Based Simulation 2004, Springer, vol 3415, pp 130–144

Epstein JM, Axtell R (1996) Growing Artificial Societies. Brookings Institution Press and MIT Press

Erev I, Roth A (1998) Predicting how people play games: reinforcement learning in experimental games with unique mixed-strategy equilibria. American Economic Review 88

Evans R (1976) Modelling the Economic Objectives of the Physician. In: Fraser R (ed) Health Economics Symposium, Queen's University and Kingston, Canada, Industrial Relations Centre, Queen's University. Kingston, Ont.

Fagiolo G, Moneta A, Windrum P (2007) A Critical Guide to Empirical Validation of Agent-Based Models in Economics: Methodologies, Procedures, and Open Problems. Computational Economics 30:195–226

Falk A, Kosfeld M (2003) It's all about Connections: Evidence on Network Formation, Institute for the Study of Labor, Discussion Paper No. 777

Fang H, Moro A (2011) Theorie of Statistical Discrimination and Affirmative Action: A Survey. In: Jess Benhabib MOJ, Bisin A (eds) Handbook of Social Economics, The Netherlands: North-Holland, pp 133–200

Flache A, Macy MW (2002) Learning dynamics in social dilemmas. Proceedings of the National Academy of Science 99:7229–72,346

Fryer Jr RG, Goeree JK, Holt CA (2005) Experience-Based Discrimination: Classroom Games. Journal of Economic Education 36(2):160–170

Gabbott M, Hogg G (1993) Choice of GP: Fact or Fiction. Journal of Management in Medicine 7(1):57–63

Gage H, Rickman N (2000) Patient Choice and Primary Care, Department of Economics, University of Surrey

Galeotti A, Goyal S, Kamphorst J (2006) Network formation with heterogeneous players. Games and Economic Behavior 54:353–372

Gaynor M (2006) What Do We Know About Competition and Quality in Health Care Markets?, Carnegie Mellon University, University of Bristol

Gifford S (2005) Limited Attention as the Bound on Rationality. Contributions to Theoretical Economics 5(1)

Gigerenzer G, Goldstein DG (1996) Reasoning the fast and Frugal Way: Models of Bounded Rationality. Psychological Review 103(4):650–699

Gilbert N (2006) When Does Social Simulation Need Cognitive Models? In: Sun R (ed) Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation, Cambridge University Press, chap 19, pp 428–433

Gilboa I, Schmeidler D (1996) Case-Based Optimization. Games and Economic Behavior 15:1–26

Goeree JK, Riedl A, Ule A (2009) In search of stars: Network formation among heterogeneous agents. Games and Economic Behavior 67:445–466

Goldberg DE (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional

Goodwin G (1998) GP Fundholding. In: Le Grand J, Mays N, Mulligan J (eds) Learning from the NHS internal market: A review of the evidence, King's Fund, London

Gosden T, F F, Kristiansen I, Sutton M, Leese B, Giuffrida A, Sergison M, Pedersen L (2000) Capitation, salary, fee-for-service and mixed systems of payment: effects on the behaviour of primary care physicians (Review). Cochrane Database of Systematic Reviews 3

Gotts NM, Izquierdo LR, Izquierdo SS, Polhill JG (2007) Transient and asymptotic dynamics of reinforcement learning in games. Games and Economic Behavior 61:259–276

Goyal S (2007) Connections. Princeton University Press

Granovetter MS (1973) The strength of weak ties. American Journal of Sociology 78:1360–1380

Gravelle H, Masiero G (2000) Quality incentives in a regulated market with imperfect information and switching costs: capitation in general practice. Journal of Health Economics 19:1067–1088

Grignon M, V P, Polton D (2002) Influence of Physician Payment Methods on the Efficiency of the Health Care System. Discussion Paper No 35, Commission on the Future of Health Care in Canada

Grytten J, Sorensen R (2001) Type of contract and supplier-induced demand for primary physicians in Norway. Journal of Health Economics

Grytten J, Sorensen R (2007) Primary physician servicesList size and primary physicians service production. Journal of Health Economics 26:721–741

Hamill L, Gilbert N (2009) Social Circles: A Simple Structure for Agent-Based Social Network Models. Journal of Artificial Societies and Social Simulation 12(2):3

Hickson G, Altemeier W, JM P (1987) Physician reimbursement by salary or fee-for-service: effect on physician practice behavior in a randomized prospective study. Pediatrics 80(3):344–350

Hole A (2008) Modelling heterogeneity in patients' preferences for the attributes of a general practitioner appointment. Journal of Health Economics 27:1078–1094

Holland J (1975) Adaptation in natural and artificial systems: An introductory analysis with application to biology, control, artificial intelligence. University of Michigan Press, Ann Arbor

Holland J, Booker L, Colombetti M, Dorigo M, Goldberg D, Forrest S, Riolo R, Smith R, Lanzi P, Stolzmann W, Wilson S (2000) What Is a Learning Classifier System? In: Lanzi P, Stolzmann W, Wilson S (eds) Learning Classifier Systems, Lecture Notes in Computer Science, vol 1813, Springer Berlin / Heidelberg, pp 3–32

Hopkins E, Posch M (2005) Attainability of boundary points under reinforcement learning. Games and Economic Behavior 53:110–125

Hummon N (2000) Utility and dynamic social networks. Social Networks 22:221–249

Hutchinson B, Birch S, Hurley J, Lomas J, Stratford-Devai F (1996) Do physician-payment mechanisms affect hospital utilisation? A study of Health Service Organisations in Ontario. Canadian Medical Association Journal 154:653–661

Irwin J, Kiczales G, Lamping J, Loingtier Jm, Lopes C, Maeda C, Mendhekar A (1997) Aspect-Oriented Programming. In: Proceedings of the European Conference on Object-Oriented Programming, vol 1241, pp 220–242

Jackson M, Rogers B (2005) The Economics of Small Worlds. Journal of the European Economic Association (Papers and Proceedings) 3(2-4):617–627

Jackson M, Watts A (2002) The Evolution of Social and Economic Networks. Journal of Economic Theory 106:265–295

Jackson M, Wolinsky A (1996) A Strategic Model of Social and Economic Networks. Journal of Economic Theory 71:44–74

Jackson MO (2008) Social and Economic Networks. Princeton University Press

Jackson MO, van den Nouweland A (2005) Strongly stable networks. Games and Economic Behavior 51:420–444

Jelovac I (2001) Physicians payment contracts, treatment decisions and diagnosis accuracy. Health Economics 10(1):9–25

Jelovac I, Marinoso BG (2003) GPs' payment contracts and their referral practice. Journal of Health Economics 22:617–635

JGAP (2011) JGAP - Java Genetic Algorithms Package, URL http://jgap.sourceforge.net/, last accessed August 2011

JGroups (2010) The JGroups project, URL http://www.jgroups.org/, last accessed June 2010

Kahnemann D, Tversky A (1979) Prospect Theory: An Analysis of Decision under Risk. Econometrica 27:263292

Karandikar R, Mookherjee D, Ray D, Vega-Redondo F (1998) Evolving Aspirations and Cooperation. Journal of Economic Theory 80:292–331

Karlsson M (2007) Quality incentives for GPs in a regulated market. Journal of Health Economics 26:699–720

Kernick D (2006) Wanted new methodologies for health service research. Is complexity theory the answer? Journal of Health Economics 23:285–390

Kirman A, Vriend N (2001a) Evolving market structure: An ACE model of price dispersion and loyalty. Journal of Economic Dynamics & Control 25:459–502

Kirman A, Vriend NJ (2001b) Evolving market structure: An ACE model of price dispersion and loyalty. Journal of Economic Dynamics and Control 25:459–502

Krasnik FH, Groenewegen P (1992) Introducing fees for services with professional uncertainty. Health Care Financing Review 14(1):107–115

Kristiansen I, Hjortdahl P (1992) The general practitioner and laboratory utilization: why does it vary? Family Practice 9:22–27

Kristiansen I, Holtedahl K (1993) The effect of the remuneration system on the general practitioners choice between surgery consultations and home visits. Journal of Epidemiology and Community Health 47:481–484

Kristiansen I, Mooney G (1993) The general practitioners use of time: is it influenced by the remuneration system? Social Science and Medicine 37:393–399

Laslier JF, Walliser B (2005) A reinforcement learning process in extensive form games. International Journal of Game Theory 33:219–227

Laslier JF, Topol R, Walliser B (2001) A Behavioral Learning Process in Games. Games and Economic Behavior 37:340–366

LeBaron B, Arthur W, Palmer R (1999) Time series properties of an artificial stock market. Journal of Economic Dynamics & Control 23:1487–1516

Lehman J, Laird J, Rosenbloom P (2003) A Gentle Introduction to Soar, an Architecture for Human Cognition, URL http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/Gentle.pdf, University of Michigan

Leombruni R, Richiardi M (2005) Why are economists sceptical about agent-based simulations? Physica A 355:103–109

Lerner C, Claxton K (1996) Modelling the Behaviour of General Practitioners. Discussion paper 116, Centre for Health Economics, University of York

Levaggi R, Rochaix L (2007) Exit, choice or loyality: Patient driven competition in primary care. Annals of Public and Cooperative Economics 78(4):501–535

Levin J (2009) The Dynamics of Collective Reputation. The BE Journal of Theoretical Economics 9(1):1–23

Lopes LL (1994) Psychology and Economics: Perspectives on Risk, Cooperation and the Marketplace. Annual Review of Psychology 45:197–227

Luce RD (1959) Individual Choice Behavior: A Theoretical Analysis. Wiley, New York

Ma C (1994) Health care payment systems: cost and quality incentives. Journal of Economics Management and Strategy 3(93-112)

Marinoso B, Jelovac I (2003) GPs' payment contracts and their referral practice. Journal of Health Economics 22(4):617–635

Markham A (1999) Knowledge Representation. Lawrence Erlbaum Associates, Mawah NJ

MASON (2010) The MASON project, URL http://cs.gmu.edu/~eclab/projects/mason, last accessed June 2007

McBride M (2006) Imperfect monitoring in communication networks. Journal of Economic Theory 126:97–119

McConnel CR, LBrue S, Macpherson DA (2006) Contemporary Labor Economics. McGraw-hill/Irwin, New York

McGuire A, Rickman N (1999) Regulating provider's reimbursement in a mixed market for healthcare. Scottish Journal of Political Economy 46(1):53–71

Milgram S, Travers J (1969) An Experimental Study of the Small World Problem. Sociometry 32(4):425–443

Minsky M (1975) A framework for the representation of knowledge. In: Winston P (ed) The pschology of computer vision, McGraw Hill, New York, pp 211–277

Minson R, Theodoropoulos GK (2004) Distributing RePast Agent-Based Simulations with HLA. Concurrency and Computation: Practice and Experience 00:1–25

Mookherjee D, Sopher B (1994) Learning Behaviour in an Experimental Matching Pennies Game. Games and Economic Behaviour 7:62–91

Mookherjee D, Sopher B (1997) Learning and Decision Costs in Experimental Constant-sum Games. Games and Economic Behaviour 19:97–132

Mooney G, Ryan M (1993) Agency in health care: getting beyond first principles. Journal of Health Economics 12(2):125–135

Napel S (2003) Aspiration adaptation in the ultimatum minigame. Games and Economic Behavior 43:86–106

Nene D (2005) A beginners guide to Dependency Injection, URL http://www.theserverside.com/news/1321158/A-beginners-guide-to-Dependency-Injection/, The Server Side online journal

Pemantle R, Skyrms B (2000) A dynamic model of network formation. Proceedings of the National Academies of Science 97:9340–9346

Pemantle R, Skyrms B (2004) Network formation by reinforcement learning: the long and medium run. Mathematical Social Sciences 48(3):315–327

Phelps E (1972) The Statistical Theory of Racisim and Sexism. American Economic Review 62:659–661

PostgreSQL Global Development Group (2010) PostgreSQL, URL http://www.postgresql.org/, last accessed September 2010

Repast (2010) The Repast project, URL http://repast.sourceforge.net/repast_3, last accessed May 2010

Richiardi M (2003) The Promises and Perils of Agent-Based Computational Economics, Laboratorio R. Revelli, Centre for Employment Studies

Robertson R, Thorlby R (2010) Patient choice, The King's Fund Briefing, 2008

Roth A, Erev I (1995) Learning in Extensive Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Run. Games and Economic Behaviour 6

Royal College of General Practitioners (2006) Profile of UK General Practitioners 20006 (www.rcgp.org.uk/pdf/ISS_INFO_01_JUL06.pdf), URL www.rcgp.org.uk/pdf/ISS_INFO_01_JUL06.pdf, last accessed March 2009

Rubinstein A (1998) Modeling Bounded Rationality. MIT Press

Rustichini A (1999) Optimal Properties of StimulusResponse Learning Models. Games and Economic Behavior 29:244–273

Ryan M (1994) Agency in health care: lessons for economists from sociologists. American Journal of Economics and Sociology 53:207–218

Sandia Labs (2010) The Jess Expert System Shell, URL http://herzberg.ca.sandia.gov/jess/, last accessed June 2010

Sarin R, Vahid F (1999) Payoff Assessments without Probabilities: A Simple Dynamic Model of Choice. Games and Economic Behavior 28:294–309

Sarin R, Vahid F (2001) Predicting How People Play Games: A Simple Dynamic Model of Choice. Games and Economic Behavior 34:104–122

Schelling T (1971) Dynamic Models of Segregation. Journal of Mathematical Sociology 1:143–186

Schuster S (2012) BRA: An Algorithm for Simulating Bounded Rational Agents. Computational Economics 39(1):51–69

Schuurmans D, Schaeffer J (1989) Representational Difficulties With Classifier Systems. In: in Proceedings Third International Conference on Genetic Algorithms, Morgan Kaufmann, pp 328–333

Scott A (2000) Economics of General Practice. In: Cuyler A, Newhouse J (eds) Handbook of Health Economics, vol 1, Elsevier Science, Amsterdam, chap 22, pp 1175–2000

Scott A (2005) For love or money? Alternative methods of paying physicians, Paper presented to conference 'Sustaining Prosperity: New Reform Opportunities for Australia', Melbourne

Scott A, Hall J (1995) Evaluating the effects of GP remuneration: problems and prospects. Health Policy 31:183–195

Sigaud O, Wilson S (2007) Learning classifier systems: A survey. Soft Computing 11:1065–1078

Simon H (1956a) A Behavioural Model of Rational Choice. In: Simon H (ed) Models of man, social and rational: mathematical essays on rational human behavior in a social setting, Wiley, New York

Simon H (1956b) Rational Choice and the Structure of the Environment. In: Simon H (ed) Models of man, social and rational: mathematical essays on rational human behavior in a social setting, Wiley, New York

Simon H (2000) Bounded Rationality: Today and Tomorrow. Mind and Society 1(1):25–39

Slikker M, van den Nouweland A (2000) Network formation models with costs for establishing links. Review of Economic Design 5:333–362

Smith JM (1982) Evolution and the Theory of Games. Cambridge University Press, Ca

Stahl DO (2000) Rule Learning in Symmetric Normal-Form Games: Theory and Evidence. Games and Economic Behavior 32:105–138

Sun R, Naveh I (2007) Social institution, cognition, and survival: A cognitive-social simulation. Mind and Society 6(2):15–42

Sun R, Slusarz P (2005) The Interaction of the Explicit and the Implicit in Skill Learning: A Dual-Process Approach. Psychological Review 112(1):159–192

Sun Corporation (2010a) Enterprise Java Beans, URL http://java.sun.com/products/ejb/, last accessed June 2010

Sun Corporation (2010b) Getting started with Java Messaging Service, URL http://java.sun.com/developer/technicalArticles/Ecommerce/jms/index.html, last accessed June 2010

Sun Corporation (2010c) Java Enterprise Edition, URL http://java.sun.com/javaee/index.jsp, last accessed June 2010

Sun Corporation (2010d) Remote Method Invocation (RMI), URL http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp, last accessed June 2010

Sutton R, Barto A (1998) Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA

Swarm (2010) The Swarm project, URL http://wiki.swarm.org, last accessed June 2007

Terracotta (2010) The Terracotta Framework, URL http://www.terracotta.org, last accessed March 2010

Tesfatsion L (2006) Agent-based computational economics: A constructive approach to economic theory. In: Tesfatsion L, Judd K (eds) Handbook of Computational Economics, 2, Elsevier, chap 16, pp 831–877

The JBoss Community (2010a) JBoss Application Server, URL http://www.jboss.org/jbossas, last accessed June 2010

The JBoss Community (2010b) JBoss Cache, URL http://www.jboss.org/jbosscache, last accessed June 2010

The Repast Symphony Project (2010) Repast Symphony, URL http://repast.sourceforge.net/, last accessed May 2010

Vanin P (2002) Network Formation in the Lab: A Pilot Experiment, Universitat Pompeu Fabra

Vick S, Scott A (1998) Agency in health care. Examining patients' preferences for attributes of the doctor-patient relationship. Journal of Health Economics 17:587–605

Watts A (2001) A dynamic model of network formation. Games and Economic Behaviour 34:331–341

Watts A (2002) Non-myopic formation of circle networks. Economic Letters 74:277–282

Watts D (2004) Six Degrees: The Science of a Connected Age

Williams S, Calnan M (1991) Key determinants of consumer satisfaction with general practice. Family Practice 8:237–242

Woodward R, Warren-Boulton F (1984) Considering the effects of financial incentives and professional ethics on 'appropriate' medical care. Journal of Health Economics 3:223–237

Young HP (1993) The Evolution of Conventions. Econometrica 61(1):57–84

Zweifel P, Breyer F, Kifmann M (2005) Health Economics. Springer, Berlin

Appendices

Appendix A

A Scalable ACE Simulation Software Framework

A.1 Introduction

This section describes the software framework with which the simulations in the previous chapters have been implemented.

The major features of the framework gsim (for 'generic simulation framework') are: An interface for setting up a model in a declarative way (e.g. objects and attributes, agent behaviour rules); an application programming interface (API) which can be used to plug model-specific programmable components; and the possibility to run many simulations simultaneously or distribute a large simulation across a cluster of computers without the need to modify any model code.

The motivation to develop this system was to find a middle way between the flexibility of a programming language, and out of the box simulation tools. It is a more specialised framework than simulation toolkits

such as Swarm (Swarm 2010), Ascape (Ascape 2010), MASON (MASON 2010) or Repast (Repast 2010), because it not only provides a simulation infrastructure (e.g., a scheduler or tools for generating graphs) and a set of libraries useful for implementing models (e.g., genetic programming or network libraries). It provides an integrated set of behaviours and learning mechanisms that can be configured and require only little programming. Another area where gsim is different from other simulation software is its approach to scalability. In domains such as Artificial Life, distribution of simulations is usually based on algorithms that distribute the landscape agents live on efficiently. Large simulations are then partitioned in a way that most communication happens locally, minimising the message traffic over the network which is the most serious bottleneck of distributed simulations (see section A.3.2.1 for more details). Only few general distribution approaches not being based on topography exist. For RePast, for example, distribution has been implemented with the Terracotta framework in the RePast Symphony project (The Repast Symphony Project 2010), (Terracotta 2010), or with the High-Level-Architecture (HLA), a specification for parallel systems (e.g. Minson and Theodoropoulos 2004; Cicirelli et al 2009). For a model to become distributed, such approaches require the additional implementation or configuration of the objects that are to be distributed over the cluster, or even a redesign of single-machine programs. gsim proposes a different method which abstracts from framework-specific programming and configuration.

This appendix describes the central components and software architecture of the framework: Section A.2 presents how models are described (sections A.2.1 to A.2.2) and behaviour specified (sections A.2.3 and A.2.6). Section A.3 describes the software architecture from a more technical and point of view, including a description of how the system is scaled up to a

distributed version (section A.3.2).

A.2 Model Representation System

A.2.1 The Frame Principle

Frames are a central concept of Artificial intelligence (AI) for knowledge representation and were first described by Minsky (Minsky 1975). It is an approach to represent classes of objects on different levels of abstraction, down to their concrete realisations as objects.

A frame can be defined as a schema that describes an entity or a class of entities by a collection of attribute-value pairs in a hierarchy of such schemata. Attributes have variable character; they provide 'slots'. The slots can take a specific value ('fillers') to describe a more concrete entity. Attributes may be thought of having any type of filler, such as number or strings, but in particular, other frames. Complex structures and relationships can be generated by nesting frames into each other (similar to object-oriented concepts).

A central feature of the frame concept is inheritance. Entities of the same type can be more or less specialised depending on their position in the hierarchy. On the higher, abstract levels of a hierarchy, frames typically specify only very general information, such as the type of slots they contain and the possible fillers for these slots. On lower, more concrete levels, slots may be filled by more specific value ranges and default values. Default attribute values describe a typical object of a class. For example, a vehicle can be characterised by having a number of wheels. A car is a certain class of vehicle, which has typically four wheels, so that using number 4 as a default filler would be a sensible choice.

While frames describe classes of entities at different levels of abstraction, actual realisations of these classes are called 'instances'. An instance describes an existing object with referral to its frame, for example 'the green car with four wheels'.

Using as an example the frame firm, figure A.1 illustrates this concept, deriving concrete firms from it.

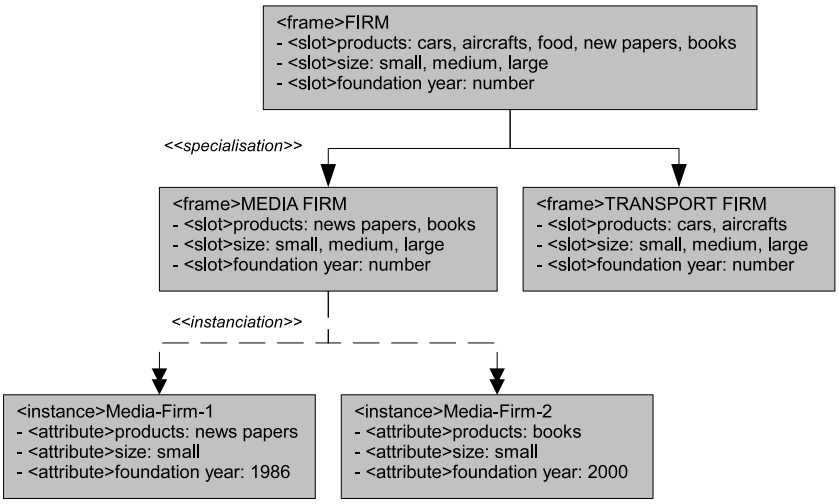


Figure A.1: Knowledge representation in gsim.

In figure A.1, the top level frame describes the possible values entities of the type firm can have. On a more concrete level, media and transport firms can be distinguished by applying default fillers. On the instance level, concrete firms are created from the default frames. In gsim, this process is labelled 'instanciation' to distinguish it from instantiation in object-oriented programming.

The frame model in gsim gsim entities are organised by lists of attributes and list of further frames. For simplification of the implementation, both types of entities are kept separate. Attributes are simple name-value pairs, and cannot contain frames.

Entities are managed in their own environment, which implements the schema hierarchy and manages value changes within that hierarchy. For example, if a default value further up in the inheritance is changed, this change is propagated down to all inheriting frames and all instances of that frame. Any entity in gsim must be created via this environment. The environment defines two special top-level frames: agent-class and object-class. Any concrete simulation model must inherit from these entities.

As in the base concept, frames describe the possible value ranges and default values. gsim makes some restrictions on the hierarchies that can be generated. Only agent classes may contain further frames; object classes are simple containers for attributes. Attributes can be of different types: numerical attributes accept any numbers; string attributes any string value; intervals define a value range by a minimum and maximum value; similarly, set attribute defines a list of allowed fillers.

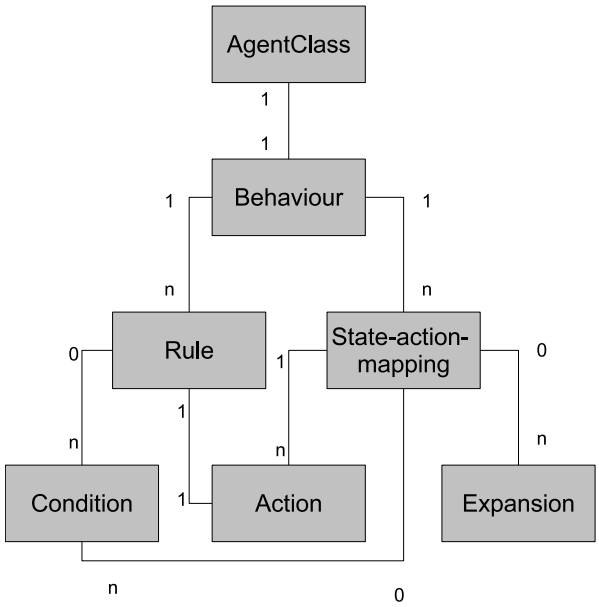


Figure A.2: The base agent frame in gsim (attributes are omitted).

Figure A.2 illustrates the agent-class. It only specifies frames for the

behaviour and adds some default attributes (e.g. the attribute that contains the operator of a condition); any other attributes and object classes would be added depending on the simulation model. The behaviour itself is given again by nested frames. A behaviour consists of a set of rules, which are composed of conditions and refer to actions. The details are given in the following section A.2.2.

During the process of instanciation, frames serve as a template for generating objects. Initially, the frame attribute defaults are used for the concrete values in the instance's attributes. Similarly, default objects may be generated. After the instances are generated, the default values may then be overridden with instance-specific values, or varied randomly (the prototype implementation provides utility classes for this).

Thus, a large range of models can be described with such a generic representation, and agent populations easily be generated and modified. Models specialise the generic classes by adding new attributes and further frames. If a simulation requires it, the addition or deletion of attributes on the instance level is possible as well.

Due to the regularity of the representation, it is possible to specify a simple language that declares the objects of a model, which at the same time is capable of specify some simple dynamics by defining rules operating on these descriptions. The following section A.2.2 presents this language, and illustrates how it is applied to generate the actual programs that execute agents in gsim.

A.2.2 Formal Description as Language

This section specifies the gsim entities and dynamics with the help of a simple language. The notation is based on the Backus-Naur (BNF) form,

a notation used to express context-free grammars. Context-free grammars are often used in Computer Science to describe the syntax of programming languages by production rules of the form $V \rightarrow w$, where V is a non terminal symbol and w is a string consisting of terminals and/or non-terminal symbols. Non-terminal symbols are enclosed in brackets '<>'. Terminal symbols are character strings (enclosed in quotation marks). The character '|' denotes a logical 'or'. An asterisk '$*$' denotes (zero or more) repetitions. The operator ":=" denotes a production rule where the expression on the left-hand side is replaced with the expression on the right-hand side. Subsequent replacements will thus resolve to a sequence of terminal symbols. The operator ':=' denotes a production of the left-hand side to the expression on the right-hand side (examples are given in section A.3.2.4). Expressions in the following paragraphs are valid also for the succeeding paragraphs, i.e. definitions in preceding sections are not redefined when referenced in later sections (e.g. character).

Common terminal symbols

```
character :: "a" | "b" | "c" | "d" | "e" | "f" | "g" |
             "h" | "i" | "j" | "k" | "l" | "m" | "n" |
             "o" | "p" | "q" | "r" | "s" | "t" | "u" |
             "v" | "w" | "x" | "y" | "z"
digit :: "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
```

Frame definition

```
<frame>                   := <entity-name> [<frame>*] <frame-list>*
                                <domain-attribute-list>*
<entity-name>             := character*
<list-name>               := character*
<frame-list>              := <list-name> "("<entity-name>")" <entity-name>*
<domain-attribute-list>   := <list-name> <domain-attribute>*
<domain-attribute>        := <domain-attribute-name>
                                <domain-attribute-type>
```

```
<domain-attribute-name>   := character*
<domain-attribute-type>   := <set-default>| <numerical-default> |
                                <numerical-interval-default> | <string-default>
<set-type>                := "Set" <filler>* <string-default-value>
<numerical-type>          :=  "Numerical" <numerical-default-value>
<numerical-interval-type> :=  "NumericalInterval"
                                <interval-default-value>
<string-type>             :=  "String" <string-default-value>
<filler>                  :=  character*
<string-default-value>    :=  character*
<numerical-default-value> :=  digit*
<interval-default-value>  :=  digit* "-" digit*
```

As described above, frames are containers for attributes and further frames. Domain attributes describe the type and default values of concrete attributes such as string- or numerical attributes. Domain attributes and contained frames are organised in lists. These lists may specify the type of object they contain by referring to the name of the entity it is allowed to contain instances of. Frames can inherit from an arbitrary number of parent frames. Inheritance can create naming conflicts, for example, lists and entities with the same name in different parents. These conflicts are not resolved, i.e. unless a particular frame is referenced, it is undefined which object is returned on the lowest level.

Instance definition

```
<instance>                :=  <entity-name> <frame> <attribute-list>*
<attribute-list>          :=  <list-name> <attribute>*
<attribute>               :=  <attribute-type> <attribute-name>
<attribute-name>          :=  character*
<attribute-type>          :=  <set-type>| <numerical-type> |
                              <numerical-interval-type> | <string-type>
<set-type>                :=  "Set" <string-value>*
<numerical-type>          :=  "Numerical" <numerical-value>
<numerical-interval-type> :=  "NumericalInterval"
                               <numerical-value>"-"<numerical-value>
```

```
<string-type>             :=  "String" <string-value>
<string-value>            :=  character*
<numerical-value>         :=  digit*
```

An instance is a concrete entity that can be generated by a frame. Any attributes or contained instances must comply with the type and domain-attribute restrictions set by its frame. An instanciation of a frame has a reference to its frame, and is accessible at any time during the life cycle of an instance.

Object class definition

```
<object-class>          :=  <entity-name> <frame> <domain-attribute-list>*
```

Object-class is a top-level object in gsim. It is a derivation of a frame that restricts the elements contained in that frame to attributes. An object class may define an arbitrary number of domain attribute lists that describe the particular entity to be modelled.

Object definition

```
<object>                :=  <entity-name> <object-class> <attribute-list>*
```

Analogous to instances, all gsim objects are derived from and refer to its defining object class.

Agent class definition

```
<agent-class>            :=  <entity-name> <frame>* <object-class-list>*
                              <domain-attribute-list>* <behaviour-class>
<object-class-list>      :=  <list-name> "("<entity-name>")" <object-class>*
<behaviour-class>        :=  <entity-name> <frame> <action-class>* <max-nodes>
                              <update-interval> <revaluation-probability>
                              <revisit-costfraction>
                              <reactive-rule-class>* <adaptive-rule-class>*
<action-class>           :=  <entity-name> <frame> <frame>* <action-java-class>
<action-java-class>      :=  character*
<reactive-rule-class>    :=  <entity-name> <frame> <condition-class>* "->"
                              <consequent-class>
<adaptive-rule-class>    :=  <entity-name> <frame> <condition-class>*
                              <expansion-class>* "->"<consequent-class>*
                              <reward-variable>
<condition-class>        :=  <entity-name> <frame> [<domain-attribute-spec>
                              <numerical-operator>
                              [domain-attribute-spec||<constant>]] |
                              [<object-class-spec> <operator>
                              [<constant> | <object-class-spec> |
                               <attribute-spec>]]
<domain-attribute-spec>  := <list-name>"/"<domain-attribute-name>"/"
                              <attribute-value>
<object-class-spec>      :=  <entity-name> <Frame> <object-class-list>"/"
                              <entity-name>"::"<list-name>"/"<domain-attribute-name>"/"
                              <attribute-value>|<object-class-list>"/"<entity-name>
<expansion-class>        :=  <list-name>"/"<domain-attribute-name> |
                              <object-class-list>"/"<entity-name>::
                              <list-name>"/"<domain-attribute-name>
<max-nodes>              :=  digit*
<update-interval>        :=  digit*
<revaluation-probability>:=  digit*
<revisit-costfraction>   :=  digit*
<reward-variable-class>  :=  <domain-attribute-spec>*
<constant-class>         :=  digit* | character*
<numerical-operator>     :=  "=" | ">" | ">=" | "<" | "<="
<operator>               :=  <numerical-operator> | "EXISTS" | "NOT EXISTS"
```

Agent-class is the top-level agent frame in gsim, and all models have to derive their agents from this frame. Agent-class specifies further frames for describing the base of agent behaviour, e.g. in conditions and actions.

Agent definition

```
<agent>              :=  <entity-name> <agent-class>* <object-list>*
                          <attribute-list>* <Behaviour>
<object-list>        :=  <list-name> <object>*
<behaviour>          :=  <entity-name> <behaviour-class> <action>* <max-nodes>
                           <update-interval> <revaluation-probability>
                          <revisit-costfraction>
                          <reactive-rule>* <adaptive-rule>*
<action>             :=  <entity-name> <action-class> <instance>* <action-java-class>
<reactive-rule>      :=  <entity-name> <reactive-rule-class> <condition>* "->"
                          <consequent>
<adaptive-rule>      :=  <entity-name> <adaptive-rule-class> <condition>*
                          <expansion>* "->"<consequent>* <reward-variable>
<condition>          :=  <entity-name> <condition-class> [<domain-spec>
                          <numerical-operator>
                          [domain-spec||<constant>]] |
                          [<object-spec> <operator>
                          [<constant> | <object-spec> | <attribute-spec>]]
<attribute-spec>     :=  <list-name>"/"<attribute-name>"/"
                         <attribute-value>
<object-spec>        :=  <entity-name> <object-class-spec> <object-list>"/"
                          <entity-name>"::"<list-name>"/"<attribute-name>"/"
                          <attribute-value>|
                          <object-list>"/"<entity-name>
<expansion>          :=  <list-name>"/"<attribute-name>
                            <list-name>"/"<entity-name>::
                            <list-name>"/"<attribute-name>
<reward-variable>    :=  <attribute-spec>*
```

Agent is the instanciation of an agent class, analogously to the frame-instance relationships described in the previous sections.

A.2.3 Agent Behaviour

A behaviour groups different types of rules, which realise the different cases of the framework described in chapter 2: Behaviour for full CBR or LCS-type learning, behaviour implementing simple RL (element adaptive-rule), and deterministic behaviour in the form of if-then rules (element reactive-rule). Technically, all behaviour is based on production rules, having zero up to an unlimited number of conditions, and one or more consequents. Conditions are simple first-order logic predicates using operators like equal or smaller, and the existence-operators exists/not exist. The BRA algorithm is applied by specifying optional 'expansion' descriptors. These descriptors specify the set of symbols and operators that constitute the propositional set $\mathcal{L}$ in definition 4, i.e. the initial condition symbols on which the rule generalisation and specialisation mechanisms work.

Conditions are patterns referencing attributes or objects of the agents. They specify object and attribute value combinations that trigger the action part of the rule, for example, 'for all objects x with attribute y greater z do ...'. Here lies the major benefit of the language specification: It is a directive that generates patterns that serve as the input for the production rule system.

Actions modify the state of the agent or initiate a conversation with other agents. Model implementations must provide an action implementation extending the framework java class gsim.engine.behaviour.SimAction. Actions may have a dynamically changing arbitrary number of arguments referring to agent objects and attributes. These arguments are specified on the frame-level. By this, it is possible to program a general action, and apply the action to a large number of unknown, dynamically changing object instances and/or attributes. For example, an action 'Sell' can implement

the selling of an object in an agent's product list. Instead of programming an action 'Sell product X', referring to a product instance, a general action for 'Sell a product' can be implemented and parameterised with the object-class that the product list holds. At runtime, the rule engine will determine the product to be sold based on the condition and pass it as actual parameter to the action.

The following paragraphs describe the gsim rule system in more detail.

Rules The core of the simulation system is the rule engine, based on the rule system Jess (Sandia Labs 2010). Depending on the agent's current state, the pattern matcher determines which rules are to be fired. The pattern matcher searches objects in the rule engine's knowledge base and finds those combinations that match the objects described in the condition part. In gsim, the knowledge base is constituted by the part of the agent's state (defined as the objects and attributes of the agent) that is referred to in the behaviour specification.

In a model specification, rules are typically specified on the frame level. The state to be evaluated is given by the terms <object-class-spec> and <domain-attribute-spec>. Pattern matching in case of an <object-class-spec> follows the rule: For all objects of type <frame-name> with attribute <attribute-name> and value <attribute-value> execute <consequent> [java-class]. Several terms of <object-class-spec> and <domain-attribute-spec> can be combined in one condition, and are interpreted as connected with a logical 'and'. During runtime, objects and attributes are bound to variables. The variable names are given by the names of the respective frame and domain-attributes. If instances or attributes of the same type are referenced several times in <object-class-spec> and <domain-attribute-spec>, they are bound to the same variable. This way it is guaranteed that con-

sequents are only triggered by concrete values of the same entity, and not arbitrary combined instances that are found in the knowledge base.

From this pattern matching and variable binding logic three characteristic cases of how and how often rules may become activated can be distinguished: In case (1) an unparameterised consequent is fired as many times as instances in the condition are matched; in case (2) an unparameterszed consequent is fired only once independent of the concrete matches; and in case (3) a parameterised consequent is matched once by binding the action parameters to the variables in the condition. The following examples illustrate these cases:

1. A simple action may be activated many times depending on the objects in a list.

```
agent-class       := Agent object-class-list-1 (object-class-1) behaviour-class-1
behaviour-class-1 := BehaviourClass1 rule-class-1
rule-class-1      := RuleClass1 condition-class-1 -> action-class-1
action-class-1    := DoSomethingClass1 executable.class.java
object-class-1    := AssetClass1 domain-attribute-list-1
condition-class-1 := ConditionClass1 object-class-list-1/object-class-1::
  domain-attribute-list-1/AssetAttribute-1 = 0
```

   This behaviour specifies a rule class that activates the rule instance of RuleClass1 any time the AssetAttribute-1 of objects in the instance list of object-class-list-1 equals 0.

2. The previous case means that an identical action is executed solely depending on the number of objects in the list. As long as the list does not always contain a singleton instance, this is probably not desirable. Usually, it will be enough (or even required) that the rule is fired only once. For this, the $EXISTS$ may be used. Using the specification declared in case (1), this behaviour becomes:

```
behaviour-class-2 := BehaviourClass2 rule-class-2
rule-class-2      := RuleClass2 condition-class-1 condition-class-2-> action-class-2
condition-class-2 := ConditionClass2 object-class-list-1/object-class-1::
  domain-attribute-list-1/AssetAttribute-1 EXISTS
```

This rule first tests whether the attribute exists and is 0. Technically, it restricts the exists-test to those attributes with value 0, which is redundant but has the effect that the rule is only fired once.

3. If there can be more than one object in a list, the model typically implies semantically that an action is activated for that particular object - e.g., 'if the product is green, sell it'. Using again the declarations given above, this case is given by the following specification:

```
behaviour-class-3 := BehaviourClass3 rule-class-3
rule-class-3      := SalesRule condition-class-3 -> action-class-2
action-class-2    := Sell object-class-1 SellAction.java
condition-class-3 := ColourCondition object-class-list-1/object-class-1::
  domain-attribute-list-1/AssetAttribute-1 = 1
```

In this example, the action has been parameterised with objects of type object-class-1 (which could, say, represent a product, and the attribute a code for a particular colour). Since the condition refers also to objects of type object-class-1, the object instance matching the condition is bound to the same variable as the object instance passed to the action SellAction. The SellAction implementation then knows which object actually matched the condition, and do something appropriately with that object (e.g., sell it).

Adaptive Rules Adaptive rules are extensions of the simple rules described in the preceding paragraph and are used to implement the different types of learning described in chapter 2. Instead of one consequent, there

are several, and the action to be executed is selected probabilistically. The condition-part remains the same as in a simple rule.

Simple RL is given if there is no condition, and at least two consequents are specified. Some action is always selected, and the reinforcement, given by <reward-variable>, is updated.

CBR is given if at least one condition is specified and at least two consequents are specified. RL then applies then only in certain situations.

CBR becomes dynamic when the <expansion> element is specified. <expansion> refers to an attribute in the agent or one of its objects. This attribute must be of type <numerical-interval> or <set-attribute> to define the value range within the state-space partitioning algorithm works.

In the current implementation, the value ranges of <expansion> elements have to be fixed at setup-time; the disadvantage of this is that, as mentioned in 2, the expansion process may operate on attribute ranges that are irrelevant (for example, an initial value range 0-1000, where only values 0-10 can occur during the simulation). This makes the implementation sensitive to the setup and the expected values. It is, however, quite easy to extend the current mechanism to a more dynamic mode which is capable of integrating new values and value ranges as they appear during a simulation, making the algorithm more robust. The idea is as follows:

- If no tree is developed at all: Simply add the root node with one value or value range: If the attribute is a set-attribute, add exactly the category. If it is numerical, construct an interval (e.g. for a value x, construct a range from $x - x/2$ to $x + x/2$, or simply from x to x).
- If more than one level is developed: Select a child of the base rule at

the deepest expanded level. Add the new category to this node, or construct a new interval if the attribute is numeric. Up to the root rule, add to all predecessors of the modified rule the new category or modify the interval in an analogous way. The state value of the modified nodes remains the same, i.e. the existence of the new value does not affect current evaluations. As the new value is connected by an 'or' to the existing terms in the condition, this can be interpreted as a null hypothesis that the new value does not influence the well-being of the agent at the current time step.

Parameters controlling the execution of the algorithm (e.g. the cost of visiting nodes, the maximum allowed number of expansions and so on) are given by <max-nodes> (parameter χ in chapter 2), <update-interval> (μ in chapter 2; ν is currently fixed with $round(\mu - \frac{1}{4}\mu)$), <revaluation-probability> (ρ in chapter 2) and <revisit-costfraction> (ζ in chapter 2). These parameters are specified only once per agent, since they specify properties of an agent's mind.

The <reward-variable> (parameter p in chapter 2) element specifies which attribute is used as a reward for the reinforcement learner. Typically, a model will modify this variable as the result of a change in the agent's state. The rule engine is responsible for mapping this value into the action reward and later select an action accordingly.

The following example describes an adaptive rule that uses the attribute player-type as the variable partitioning the state space, and the current payoff of the player as the reward variable:

```
agent-class       := PDAgent player-list (player-name) attribute-list behaviour-class
player-name       := PDPlayer domain-attribute-list-1
behaviour-class   := PDBehaviourClass pd-rule-class
```

```
pd-rule-class     := PDAdaptiveClass expansion-class-1 -> action-class-1, action-class-2
action-class-1    := Cooperate models.pd.Defect
action-class-2    := Defect models.pd.Cooperate
expansion-class-1 := ExpansionClass1 player-list/player-name::
  domain-attribute-list-1/NameAttribute-1 = A|B|C|D attribute-list/payoff
```

The rule engine will use the four possible attribute values A B C or D to construct the initial rule: 'player-type= (A or B or C or D)', and create specialisations during the simulation, e.g. player-type= (A or B)'. Depending on the actual reward structure, selection probabilities will vary for different specialisations as described in detail in chapter 2.

The expansion mechanism takes a path in which identical descriptors are generated at different sections of the tree (this might be the case if there are at least two attributes). Then the order at which the rules get activated is random. Whichever rule is fired first is executed; the execution of any other rules in the same time step is suppressed.

A.2.4 Agent Communication

Agent interaction is essential to ABM. In gsim, interactions require explicit communication via messages. An agent A wanting to interact with agent B sends a message with some content, which may trigger some activity in agent B. Agent B sends a message back. In a minimal communication act, this message ends the interaction, but also might trigger further actions in agent A, who may continue talking to B and so on. This is called a communication protocol. Each protocol is executed within a single time step. Agents act without delay; that is, when the message is delivered, the receiving agent reacts immediately.

As in any discrete simulation engine, gsim agents act sequentially. This means that if agent B's turn is after agent A, agent B might change its state

during the communication, but before it is actually its turn in the normal execution order. gsim does not request or provide any rules that handle the case where this conflicts with the model logic. The modeller has to ensure in the implementation that such side-effects are avoided or controlled. During implementation of the models it was usually enough to apply state- and reward updates before a cycle of activity. gsim's scheduling method allows to partition agents into different roles which are executed separately (the roles are given by extending agent classes; the actions of each extension is executed in its own cycle). This way, the modeller can configure quite atomic units of work to ensure a correct order of actions and corresponding state updates.

As all agent activities are initiated by a rule consequent, communication protocols can technically be seen as an action. The protocol is provided by the modeller, extending some classes of the framework, and specifying it as the consequent of a rule. gsim then takes care that the messages are delivered appropriately to the involved agents.

A.2.5 Other Components

gsim provides the possibility to integrate custom procedures in the form of special agents, which are called Application Agents (e.g. for data collection, or broadcasting messages to the whole agent population). Application Agents have access to the complete model state, and are called before and after the execution of a time step. They are, in principle, helper classes and thus not represented as frames and instances, but are simple java classes that can be extended by the modeller.

A.2.6 Interfaces

In the previous section, gsim was described in terms of a context-free grammar. Based on examples. This section shortly describes two actual interfaces that have been built on this idea in the prototype. Modellers can use either the java application programming interface (API) or XML files to specify a model. It is furthermore required to use and extend certain classes of the framework to be a runnable gsim application, e.g. action implementations. The full API is available at http://www.stephan-schuster.net/gsim-docs/api-docs/index.html. The XML schema is available at http://www.stephan-schuster.net/gsim-docs/schema/model.xsd.

An XML example Listing A.1 shows how an agent definition is set up using the XML interface. It is part of a prisoner's dilemma model in which agents have several visible tags, and may learn to discriminate based on this information.

Listing A.1: XML example

```
<agent name="PDAgent" extends="GameAgent" >

  <attribute-lists>
    <list name="properties">
      <Set name="own-tag" default="BLUE" >
      <value>BLUE</value>
      <value>GREEN</value>
    </Set>
    </list>
    <list name="internal-state">
      <Set name="current-strategy" default="">
        <value>Cooperate</value>
        <value>Defect</value>
      </Set>
    <Numerical name="payoff" default="0" />
    </list>
  </attribute-lists>
```

```
  <object-lists>
    <list name="current" type="Player" />
    <list name="known-tags" type="Tag" />
  </object-lists>

  <available-actions>
    <action name="Defect" />
    <action name="Cooperate" />
  </available-actions>

  <rl-nodes>
    <rl-node name="RL-1">
      <condition-nodes>
        <condition-node param="known-tags/Tag::description/
            characteristic" op="EQ" value="current/Player::list/
            colour" />
        <expand-node param="known-tags/Tag::description/
            characteristic" />
      </condition-nodes>
      <action-nodes ref="Defect,Cooperate" />
      <default-reward value="0.5" />
      <selector value="softmax" />
      <function variable="internal-state/payoff" update-lag="1"
          alpha="0.08" />
      <discount value="0.05" />
      <averaging discount="0.1" />
    </rl-node>
  </rl-nodes>
</agent>

<objects>
  <object name="Tag">
    <list name="description">
        <Set name="characteristic" default="BLUE" >
        <value>BLUE</value>
        <value>GREEN</value>
        <value>YELLOW</value>
      </Set>
    </list>
  </object>

  <object name="Player">
    <list name="list">
      <String name="name" default="stephan" />
        <Set name="colour" default="BLUE" >
```

```
        <value>BLUE</value>
        <value>GREEN</value>
      </Set>
    </list>
   </object>
</objects>

<actions>
   <action name="Defect" class="models.pd.Defection" />
   <action name="Cooperate" class="models.pd.Cooperation" />
</actions>

<system-agents>
  <system-agent name="PairingGenerator" class="gsim.sim.agent.
      gameagents.FullInformationPairingGenerator" />
</system-agents>
```

Listing A.1 describes an agent with several attributes in the <attributes> section and objects in the <objects> section. An agent is described by its name and colour. It has two object lists. The list 'current' is a singleton list containing the current other player (the selection of which is implemented in the 'PairingGenerator' ApplicationAgent which responsible for matching players). The known-tags list is also singleton, and makes the tags defined for the simulation known to the agent. Its attributes can then be referenced in the behaviour part as variables. The java implementation of the actions is given in the concluding <actions> tag. The agent has an adaptive rule. The three attributes of the expansion element <Tag> can be expanded into six different combinations. Following the variable binding rules described in the previous section, of the different rules that may exist during run-time, only that rule is activated where the characteristic attribute equals the colour of the current player. The reward variable is given by the attribute 'internal-state/payoff'. The reward is updated in a separate cycle: PDAgent has is an extension of 'GameAgent', the reward is updated in the GameAgent role (not displayed here).

API Listing A.2 indicates how the example lines in listing A.1 would be implemented programmatically.

Listing A.2: API example

```
Core core = gsim.core.CoreFactory.getInstance().createCore();

//The environment is responsible for maintaining the agent- and
   object hierarchy
DefinitionEnvironment env = core.create("PrisonersDilemma", new
   java.util.HashMap());

//define the game agent here
AgentClassIF gameAgent = env.createAgentClass("GameAgent");

//[...]

//PDAgent inherits from GameAgent
AgentClassIF pd = env.createAgentClass("PDAgent", "GameAgent");

//define the object-lists for respective types
ObjectClass tag = env.getObjectClass("Tag");
pd.defineObjectList("known-tags", tag);

//[...]

//add some objects and attributes
DomainAttribute payoff =
  new DomainAttribute("payoff", AttributeConstants.NUMERICAL);
payoff.setDefault("0");
pd.addAttribute("internal-state", payoff);

DomainAttribute ownTag = new DomainAttribute("own-tag",
   AttributeConstants.SET);
ownTag.addFiller("GREEN");
ownTag.addFiller("BLUE");
ownTag.setDefault("BLUE");
pd.addAttribute("properties", domainAtt);

//[...]

//Define behaviour
BehaviourIF behaviour = pd.getBehaviour();
RLActionNodeIF rl   = b.createRLActionNode("RL1");
```

```
ConditionIF condition = rl.createCondition("known-tags/Tag::
    description/characteristic", "=",
    "current/Player::list/colour");
ExpansionIF expansion = rl.createExpansion("known-tags/Tag::
    description/characteristic");

ActionIF consequent1 = b.createAction("Cooperate", "models.pd.
    Cooperate");
ActionIF consequent2 = b.createAction("Defect",
    "models.pd.Defect");

rl.addOrSetCondition(condition);
rl.addOrSetExpansion(expansion);
rl.addOrSetConsequent(consequent1);
rl.addOrSetConsequent(consequent2);

behaviour.addOrSetRLActionNode(rl);

pd.setBehaviour(b);

//define system-level objects, e.g.:
env.addApplicationAgent("Matcher", "gsim.sim.agent.gameagents.
    FullInformationPairingGenerator");
```

Here, the main idea is that the attributes (which are simple name-value pairs) of the agent are referenced by strings defining where they are located. The behaviour is then composed by passing these specifications. There is no direct programming model, as, say condition.setLeftHandSide(tagObject), etc. The gsim rule parser translates the specifications into an executable Jess program.

A.3 Software Architecture

The main logic of gsim is implemented in the representation system. It provides the classes and interfaces necessary to build and run a model. The representation system itself is part of larger software architecture that provides the infrastructure for running simulations (section A.3.1.1), and its

extension to a distributed simulation system for handling large simulations (section A.3.2).

A.3.1 Base system

The base system describes the standalone software environment for running gsim models.

A.3.1.1 Architecture and Design

Three layers can be distinguished. Here, the basic responsibilities and functionality of each layer is shortly described.

Access layer This layer serves as the entry point and connects the modeller with the definition and simulation layer. The modeller defines and creates models either using the gsim API, or via the XML interface. After defining the model in the environment, the model can be simulated. The reference to the SimulationManager component is obtained via the gsim API. The SimulationManager provides control method to start, stop, pause or resume simulations. It is also possible to register event listeners that handle events like the end of a simulation. Furthermore, the SimulationManager component is used to access the current state of a simulation (i.e. all agents at time t).

Definition layer This layer contains the implementation of the representation system as described in section A.2. All frames and instances are maintained in their own environment. The environment handles relationships and inheritance. For example, if an agent-class is modified in the API, the environment propagates the changes to all subclasses and instances of this frame.

Simulation layer This layer transforms the data structures of the representation system into runnable code. The most important aspect of this is the translation of the declarative structures into a rule-based program. Furthermore, it contains the scheduler which executes the simulation in discrete time steps. The SimulationContainer component provides a possibility to repeat a model several times. For this it creates the specified number of model instances and schedulers (up to a maximum number of parallel threads) and queues the remaining instances. It notifies the SimulationManager after the execution of the model has finished. It is possible to partition a large simulation over a number of delegates running in their own threads to speed up execution.

By default, gsim uses a database to store simulation data. Accessing the storage is manged by data handler classes. The modeller configures the data source, and provides extension classes of the data handlers that insert the data into the database. The framework calls these handlers with the configured database connection. It is also possible not to use a database.

Figure A.3 illustrates the components.

A.3.1.2 Implementation

The standalone system is implemented in the Java Programming language, version 1.6. The core is the behaviour system that generates executable code from the rule descriptions, for which the rule engine Jess (Sandia Labs 2010) is used. The standalone system does by itself not require a database. For the model implementations of this thesis, PostgreSQL has been used.

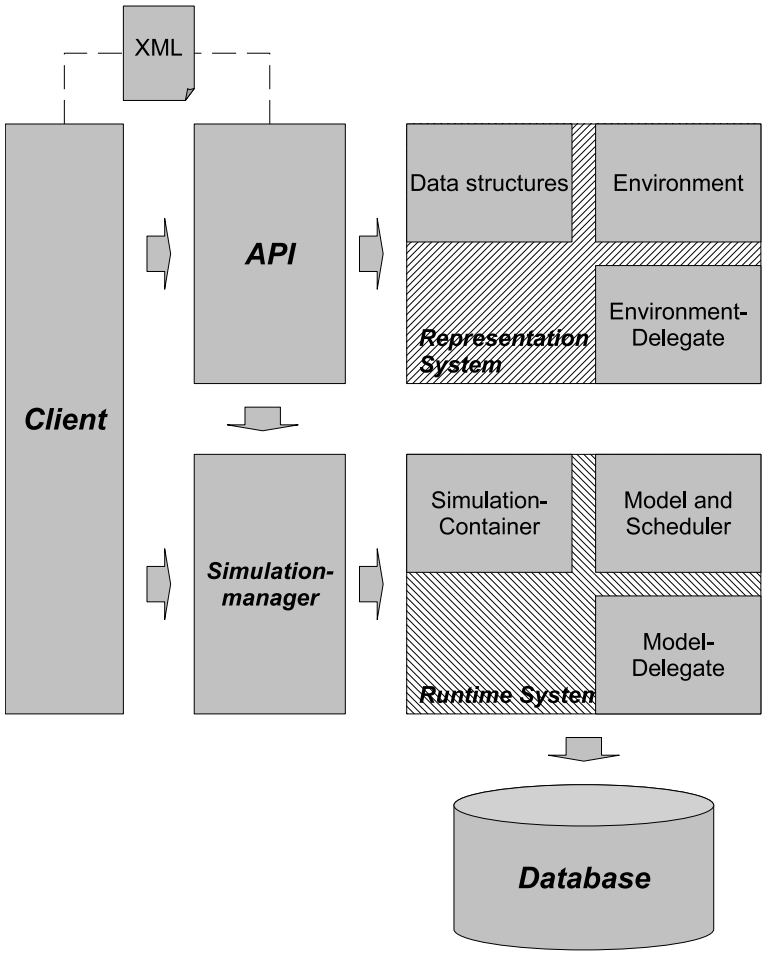


Figure A.3: gsim base architecture. Arrows denote both flows of control and object references.

A.3.2 Distributed system

A.3.2.1 Basic Design Questions

Scaling up a standalone environment raises a number of questions and implications usually not relevant for standalone social simulation systems:

Synchronisation Social simulations are typically run sequentially and synchronously. This means that exactly one agent is executed at a time, and each agent has at any time the same information about its own and

its environment's state. Remaining with this mode of execution limits the benefits of a distributed execution of a model, since all nodes in a cluster have to be synchronised, and performance might even degrade because of the coordination overhead. In some types of models, it is possible to proceed with different parts of the simulation in their own time, with no or only eventual synchronisation. This approach is useful if the agents are mostly independent and can be executed without much influence on other agents, or if clusters of interdependent agents can be identified and be separated on distinct nodes. Such concepts have been applied in distributed Artificial Life simulations where agents are located on a two-dimensional gird, and often act isolated and communicate rarely. It seems unsuitable for many social simulations, since social systems inherently require communication, interaction and shared information. Moreover, social simulations do not always require a geographical environment.

Being a general framework, distributed gsim does not provide special synchronisation algorithms for asynchronous execution. The only fundamental assumptions it builds on are (1) agents share the same time, and (2) that the perceivable environment state is identical for all agents. The framework guarantees that these conditions are satisfied at any time during the simulation. Synchronisation itself happens indirectly by configuring how messages are sent through the system, which is described in the following paragraph.

Messaging The synchronisation of the overall state of a simulation has consequences for the communication of agents in a distributed system. Since gsim explicitly makes no assumptions about the distribution of agents over the physical nodes, most agent-to-agent communication travels over the network. One option to implement communication is (1) to send many small

messages (e.g. for every single agent, messages are transmitted immediately over the network to the receiver) and achieve a high degree of parallelism. The disadvantage is a potential message overload and nodes becoming too busy just processing messages. The system could even become slower than standalone systems because the time to process messages exceeds the benefits of parallel processing. In case of the other extreme (2), all messages produced on one node could be bundled, so that nodes communicate across the network, but not single agents. This approach minimises communication, but also limits the benefits of parallel execution since most agents located on one node have to wait for all agents on other nodes, even for those who are not directly interacting with them.

The gsim approach is a solution between these extremes. Messages between agents are bundled. The modeller can configure how many messages are collected before being sent off, or switch the mechanism off. In the latter case, all messages of the node are collected and sent over the network only after all agents in the node have (case (2)). Case (1) is achieved by setting the bundling threshold to 1, which results in each agent message being immediately dispatched to the receiver.

In any distributed system, messages can get lost; servers break down or similar network failures occur. gsim does not provide a recovery mechanism, and also no built-in security to prevent failures due to communication or computation overload. Only some basic configuration parameters allow to control the workload, for example, the maximum number of concurrently running simulations, and the size of simulation partitions. An optimised configuration depends on the cluster the framework is running on, and has to be tuned by the modeller. As an illustration, on a cluster with two nodes with Pentium IV processors, a restriction to 30 parallel small simulations was found to be a reasonable upper limit.

Data collection Data collection in standalone applications typically iterates over agents in the agent container, computes statistics of interest and displays it either to the user, or stores it in files or a database. In a distributed system, this data has to be collected from different nodes and sent over the network before computations are possible. This may lead to performance or memory problems for large data sets. Using a database, gsim provides a configurable caching mechanism that delegates the computation of statistics to a separate thread or even node. Each node stores its current state in a database table, while a dedicated thread reads the data and can then do computations on it. This is not visible to users, so that no particular attention has to be paid whether data of only a few dozens or several thousand agents is collected.

A.3.2.2 Architecture and Design

In the distributed version, the components of the standalone simulation system are replicated over a cluster of servers. Some central services exist uniquely in the cluster: The simulation clock, a resource manager controlling the number of parallel executing simulations, and a central registry for configuration entries and the IP addresses of the servers currently available in the cluster. The nodes of the cluster are coordinated by a mixture of direct remote procedure calls and messages where parallelization is important.

Environment and model container objects act as master. Instead of handling agents themselves, they delegate calls to the appropriate delegate, either directly, or by sending a message to the cluster:

– The Environment master receives a request from the access layer to create a number of agents, and the number of delegates to be created.

The master then creates the delegates and allocates the same number of agents to each of them.

– When a simulation is started, the SimulationContainer receives a reference to the Environment master and creates one or more (if more than one model run is requested) master models.

– The Model master holds a reference to the Environment master. It creates the runnable agents and distributes them to number of delegates defined by a partition size parameter.

In general, gsim uses direct (remote) references and synchronous calls where possible and asynchronous messaging only during runtime. In the setup stage, most communication happens directly. Furthermore, some administrative tasks during a run can be calls by reference. Fetching the simulation state, for example, happens by (remote) referencing the Model delegates on the different nodes. To prevent too heavy memory usage, the state can be loaded in chunks. Sometimes, remote calls are not feasible anymore. For example, data collection for a very large simulation is time intensive and should be forked into a separate thread on a dynamically chosen node in the cluster (otherwise the simulation proceeds very slowly just for computing statistical information). Communication with this process happens via messages over a special channel, as it is not known where the process is located.

Asynchronous processing is used for the following cases:

– Coordinating the several master and slaves: Time in the cluster proceeds synchronised. The Model master is responsible for assuring that a whole model instance is proceeding at the same time. The Model

master receives a signal from the global clock and sends a message to its delegates requesting the execution of the next step. Each delegate sends back a finished-message after executing all its agents.

– Agent-to-agent communication: Agent-to-agent communication follows a configurable protocol to achieve a compromise between message processing load and serialisation of action sequences. More precisely, the protocol follows the following steps: An agent starting a conversation sends a starting message. The model delegate collects these messages until the bundling threshold is reached or all agents have produced their messages. The delegate then sends the messages to the cluster. All nodes in the cluster receive the message bundle and filter out those messages that address agents located at them. The receiving delegate then immediately executes the agents' responses and collects them. When the threshold is reached, or at latest after the last agent has responded, the messages are again distributed over the network. The response message content may be null when the receiver agent ends the conversation or contain an answer if the protocol consists of several steps. In the latter case, the sending of messages continues until all conversations are ended (by sending null content). This usually also ends the execution of a step.

– Accessing the current state: The Model master is accessible by system agents supplied by the modeller. Several methods to enquire about the current state exist, for example, retrieving all agents in a simulation. To access the global state, the master sends a state-request message to all delegates and waits until it has received the expected number of answers. The result can then be returned to the requesting client.

– Data caching: For large simulations, the model delegates dump their

current state into a database cache. Whenever data handlers are called to process the simulation state, the Model master sends a message containing the data handler and the reference to the cache entries to a message receiver, which does the actual computation. At the same time, the master can proceed with the simulation.

A.3.2.3 Implementation

A central feature of the system is that it builds on available standards and open source frameworks. In particular, it uses the Java Enterprise Edition (JEE) specification (Sun Corporation 2010c) that has become a widely used standard for distributed applications using the programming language Java. JEE is the general notion for several sub-specifications - for example, the Servlet API for building dynamic web-applications; a messaging specification for synchronous and asynchronous communication over the network; or Enterprise Java Beans for (synchronous) remote procedure calls. Several open-source software projects implement these standards and are provided in an application server. For distributed gsim, the famous open source application server JBoss (The JBoss Community 2010a) was chosen. While earlier JEE versions had the reputation of being very complex and difficult to manage, in recent years substantial modifications have been introduced simplifying development significantly. Software engineering principles like Aspect Oriented Programming or dependency injection (e.g. Irwin et al 1997; Nene 2005) follow a philosophy of inversion of control. The result is that much infrastructure and low level work that formerly had to be implemented or configured by the developer is now provided by the application server provider. As a consequence of these developments, it has become much easier and straightforward to extend a single-machine software to a distributed system with minimal effort for developing the necessary infras-

tructure.

gsim is a very pragmatic approach. The idea was to minimise own developments and to find a way of utilising existing open-source software to the largest extent possible. This allowed the researcher to implement the system alone. The following paragraphs shortly describe the components provided by the application server, and the major features and steps that have to be implemented to extend from the standalone to a distributed version of the software.

Standards and technologies used

Enterprise Java Beans (Sun Corporation 2010a; EJB) EJB is a technology based on the Java standard for remote method invocation, using the Remote Method Invocation protocol (Sun Corporation 2010d; RMI). In RMI, code is divided into server code and client code. A special compiler generates stubs and skeleton classes that handle the receiving and dispatching of method calls depending on the underlying network protocol. EJB builds on this base technology, simplifies its use, and provides additional features and services that are common to distributed applications, such as transaction handling or session management (e.g. by passivating or removing objects).

Java Messaging Service (Sun Corporation 2010b; JMS) JMS is a specification of a messaging middleware. JMS can be used for both synchronous and asynchronous messaging; in gsim the asynchronous mode is the most important, since the major motivation is to achieve parallel execution and loose coupling of server nodes. At the core of a JMS system is a server to which message producers and message consumers connect, i.e. it is a centralised system where the server handles the receiving and distribu-

tion of messages. Communication can be point-to-point or topic based. In the point-to-point model, one message producer sends messages to a queue, which are consumed by exactly one listener. In the topic model, several consumers listen to incoming messages. In gsim, mostly the topic approach is used. For example, the central clock service publishes step messages to which all running models react, or the Model master sends coordination messages to a single topic to which all delegates are connected. Because JMS is centralised, scalability is limited. The more messages are produced (e.g. by adding new nodes and/or running more models in parallel), the higher the load on the server, and the system may slow down or even run out of memory. gsim uses JBoss messaging, a clustered JMS server that distributes the load over all participating nodes, so that the messaging load may increase in parallel with the number of servers.

JBoss Cache (The JBoss Community 2010b) is a proprietary service based on JGroups (JGroups 2010). JGroups is a toolkit for multicast communication and can be used to create groups of processes on different computers that coordinate by sending messages. It provides various features such as group member detection and membership events such as notification about joined, left or crashed members and similar services. A major advantage of the toolkit is that the node names or IP addresses need not be known in advance. JBoss Cache uses this protocol for a distributed caching mechanism. The cache is, in principle, a tree structure that can be discovered in the local network, and into which information can be stored by the group members. In gsim, this service is used to register new nodes as they enter the network, unregister them when they are killed, remark their current load (used by gsim to distribute the workload), and to store related cluster-wide information.

PostgreSQL (PostgreSQL Global Development Group 2010) By default, a PostgreSQL database system is used to store persistent data. The JBoss application server requires a database for JMS. gsim also stores data relevant for simulations in the database, for example, information for handling the scheduling of simulations, caching the simulation state, etc. Moreover, models typically store their data into a database, for which the same database can be used.

Extension of the base architecture

The main idea of gsim is to provide the same API to the modeller independent of the standalone or clustered mode. Where the system runs is configured by a single parameter. Of course, if the simulation is very large, the modeller has to take this into account when designing data access methods, how to parameterise parallel execution and so on.

From an implementation point of view, different implementations for the environment, simulation container and model containers, as well for the messaging component are provided. Their realisation is now based on EJB and JMS technologies.

The main difference exists with respect to the management of a distributed simulation:

The cluster consists of n nodes. In each node, the same version of gsim is deployed. New nodes may be added dynamically. Removal, however, is more critical when simulations are running as gsim does not provide a failover mechanism for running simulations. Messages or objects in the cluster may get lost and prevent the finishing of simulations.

There is one central service that controls the scheduling of simulations

by setting an upper limit of parallel running models. It also provides the simulation clock (a service that issues messages at a certain interval). gsim server jobs, and their dependent objects (environment, simulation containers, model masters and delegates), are controlled by the server. If the user does not explicitly destroy objects created by him on the server, the server does this after a specified idle-timeout.

Simulations run asynchronously and autonomously in the server. The user sends a simulation model as a batch job and may disconnect. The server executes the simulation on the modeller's behalf. Reconnect to access or control the simulation is, therefore, also handled via JMS, since the references to the remote objects on the server are lost once the user disconnects.

Figure A.4 illustrates the architecture of the distributed system. To the environment and model components, delegates are added. The delegates are distributed over the available nodes in the cluster. Communication and control are mediated via the JMS server. The client API may then be located on a different computer.

A.3.2.4 Examples

The previous section made clear that perfect parallelization is difficult to achieve with the minimal (and restrictive) assumptions gsim makes about the location and communication structures of the agents. A messaging procedure has been presented that compromises between messaging overhead and maximal parallelization. Comparing simulations in distributed and standalone mode showed that the distributed version outperforms the desktop application at any stage, and that computing time increases at a flatter rate as the number of agents grows.

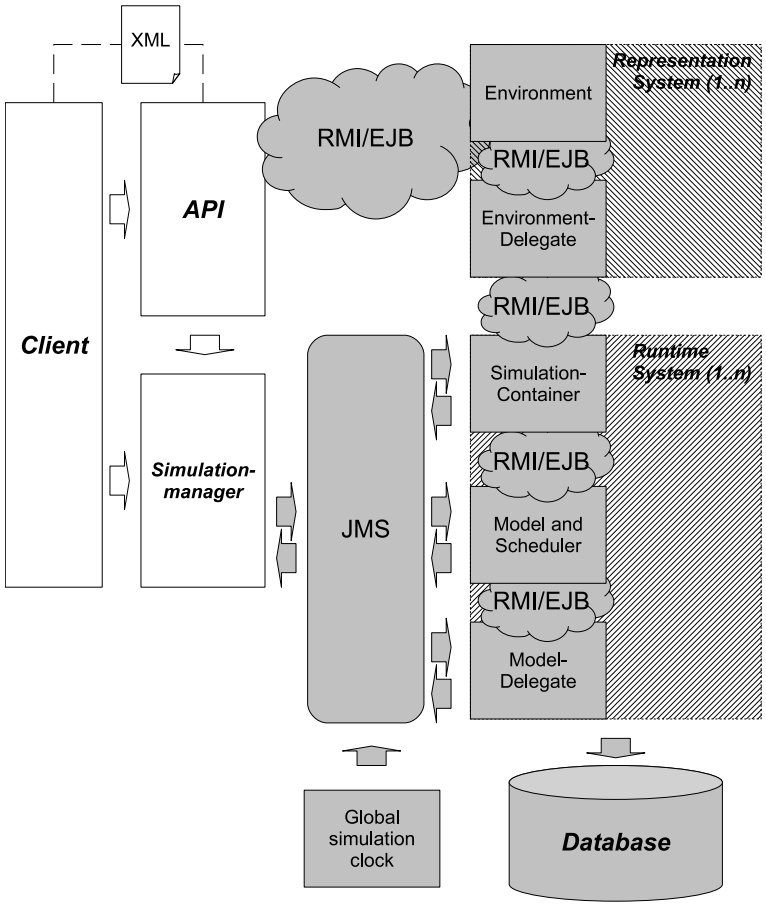


Figure A.4: Distributed architecture. White areas mark client components, gray server components. Server side components are distributed over n nodes, splitting representation and runtime over possibly n physical locations. Communication happens indirectly via messaging (JMS), or directly via remote method invocations (RMI). RMI connections typically represent object references, while JMS connections flows of control (arrows).

Figure A.5 illustrates this with a toy model with minimal communication. In the model, agents meet other agents and play a prisoner's dilemma. The communication act consists of sending each time step n messages from a central coordinator agent to n agents telling each single agent with which player they interact. Action happens in an isolated way, i.e. there is no agent-to-agent communication.

An example for a more complex simulation, including the sending of

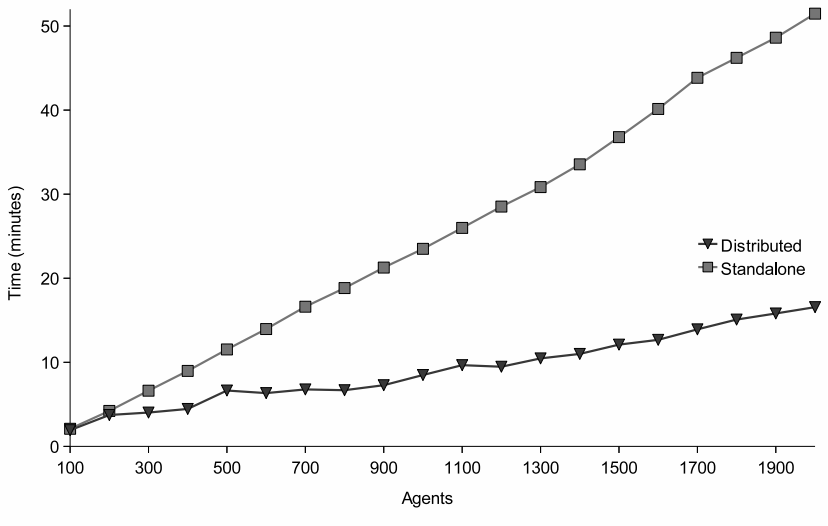


Figure A.5: Simulation examples I. In this example, the number of servers is constant as the number of agents increases.

many messages is the GP model of chapter 5. First, a model with over 3000 complex agents was hardly possible to run on a single machine with 2 GB of working memory. The results of chapter 5 were obtained (depending on availability) with up to five nodes. Figure A.6 shows some more comparative example runs with 1000 patient agents on one to four nodes. It shows first an increase in performance as the second and third nodes are added. However, the benefit of the fourth nodes diminishes. A likely reason for this is the relatively small number of agents, so that additional communication begins to outweigh the benefits of further load distribution.

The game simulations of chapters 3 and 4 represent a different use case of the system. The number of agents was very small (20 at maximum). Distributing the agents of single simulations on the cluster would not speed up the simulations (communication overhead), but the distributed version was used to execute the numerous required repetitions in parallel. In this scenario, agent communication remains local as in the standalone system, but coordination of the repetitions is realised over the network (the Simu-

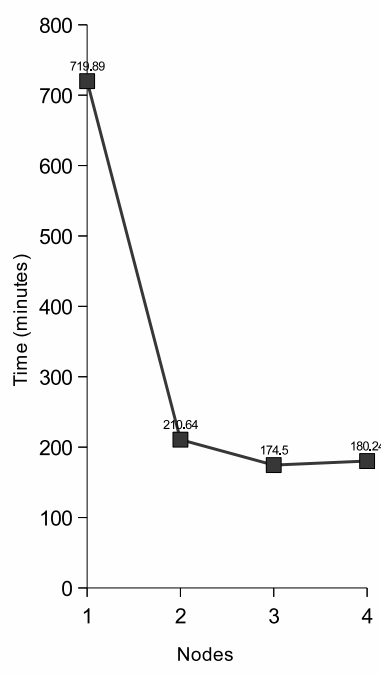


Figure A.6: Simulation examples II. In this example, the number of agent is held constant, and servers added.

lationContainer creates several Model masters on different nodes), speeding up the simulation process.

A.4 Conclusion

This appendix described the architecture and implementation of a software framework for simulating agent-based models. The framework was used to implement all models described in this thesis. The location of the source code of the framework as well as the models is listed in appendix D.

At its core is the implementation of the BRA algorithm described in chapter 2. Models using individual learning methods can easily build on this mechanism to implement deterministic behaviour, simple RL, and complex rule learning.

The software provides also the technical infrastructure to scale the system up, thus enabling the simulation of many thousand agents in reasonable time scales. It provides an API that abstracts from the fact whether the software is run in distributed or standalone mode. This makes it relatively easy to transform simple models into large-scale simulations.

These two features distinguish the gsim approach from most available modelling frameworks (e.g. Repast 2010; Ascape 2010; Cioffi-Revilla et al 2004), which often require both the implementation of behaviour strategies as well as a re-implementation of parts of the model to distribute it (e.g. Cicirelli et al 2009; Minson and Theodoropoulos 2004).

Appendix B

Details of the Statistical Discrimination Model

The following tables show some summary measures for the simulation runs of model variants I and II (chapter 2, section 3.5.2).

Table B.1: Average discrimination in model variant I. Discrimination is defined as the difference between employment levels of the high and low employment group. Each row represents averages of 5 simulation runs.

$\delta_{f(\theta)}$	avg. discrimination	st. deviation	maximum	minimum
0.0789	0.0587	0.0596	0.1603	0.0130
0.1275	0.0352	0.0236	0.0626	0.0071
0.1571	0.0452	0.0441	0.1031	0.0013
0.1681	0.0393	0.0249	0.0687	0.0031
0.1762	0.0829	0.0514	0.1367	0.0239
0.2192	0.0495	0.0324	0.0849	0.0078
0.2301	0.0380	0.0406	0.1086	0.0077
0.2317	0.0510	0.0535	0.1374	0.0005
0.2339	0.0356	0.0158	0.0600	0.0225
0.2435	0.0472	0.0401	0.1079	0.0145
0.2559	0.0493	0.0193	0.0703	0.0184
0.2610	0.0451	0.0233	0.0710	0.0098
0.3064	0.0592	0.0421	0.1285	0.0140
0.3119	0.0449	0.0157	0.0657	0.0272
0.3305	0.0433	0.0661	0.1604	0.0044
0.3567	0.0461	0.0157	0.0701	0.0280
0.3610	0.0573	0.0228	0.0772	0.0193
0.3708	0.0267	0.0243	0.0692	0.0108
0.3855	0.0529	0.0386	0.1121	0.0199
0.4076	0.0444	0.0320	0.0817	0.0073
0.4191	0.0476	0.0279	0.0674	0.0005
0.4337	0.0299	0.0124	0.0446	0.0137
0.4469	0.0502	0.0335	0.0933	0.0092
0.4558	0.0330	0.0308	0.0862	0.0113
0.4604	0.0714	0.0565	0.1533	0.0042
0.4767	0.0189	0.0214	0.0555	0.0008
0.4839	0.0450	0.0124	0.0595	0.0272
0.5245	0.0246	0.0246	0.0638	0.0024
0.5348	0.0306	0.0242	0.0687	0.0055
0.5465	0.0741	0.0612	0.1466	0.0108
0.5647	0.0261	0.0156	0.0476	0.0098
0.5679	0.0421	0.0305	0.0713	0.0018
0.5773	0.0235	0.0188	0.0564	0.0084
0.5827	0.0260	0.0287	0.0751	0.0046
0.5996	0.0506	0.0256	0.0893	0.0201
0.6188	0.0542	0.0421	0.1212	0.0114
0.6191	0.0259	0.0141	0.0451	0.0068
0.6233	0.0634	0.0475	0.1189	0.0105
0.6505	0.0296	0.0141	0.0480	0.0106
0.6817	0.0589	0.0316	0.0928	0.0209
0.6836	0.0373	0.0242	0.0770	0.0125
0.6991	0.0314	0.0300	0.0636	0.0014
0.7021	0.0477	0.0211	0.0626	0.0113
0.7066	0.0291	0.0151	0.0532	0.0165
0.7667	0.0443	0.0496	0.1320	0.0154
0.7964	0.0360	0.0340	0.0796	0.0056
0.8396	0.0419	0.0362	0.0847	0.0018
0.8561	0.0294	0.0228	0.0548	0.0023
0.9041	0.0572	0.0398	0.1164	0.0057
0.9314	0.0352	0.0198	0.0545	0.0096

Table B.2: Average discrimination in model variant II. Discrimination is defined as the difference between employment levels of the high and low employment group. Each row represents averages of 5 simulation runs.

$\delta_{f(\theta)}$	avg. discrimination	st. deviation	maximum	minimum
0.1542	0.0263	0.0244	0.0637	0.0035
0.1585	0.0654	0.1112	0.2622	0.0003
0.1933	0.0247	0.0125	0.0426	0.0090
0.2738	0.0757	0.0984	0.2484	0.0096
0.2756	0.0301	0.0244	0.0622	0.0119
0.2865	0.0296	0.0242	0.0667	0.0113
0.3075	0.0410	0.0332	0.0848	0.0067
0.3484	0.0386	0.0229	0.0632	0.0125
0.3554	0.0625	0.0545	0.1423	0.0031
0.3716	0.0372	0.0473	0.1213	0.0076
0.3885	0.0505	0.0368	0.0901	0.0016
0.4104	0.0563	0.0701	0.1776	0.0073
0.4139	0.0781	0.0796	0.1984	0.0075
0.4276	0.0343	0.0387	0.0787	0.0007
0.4339	0.0391	0.0383	0.0966	0.0029
0.4495	0.0186	0.0080	0.0259	0.0066
0.4572	0.0515	0.0422	0.1231	0.0167
0.4672	0.0521	0.0460	0.1271	0.0180
0.4713	0.0386	0.0225	0.0650	0.0132
0.4782	0.0379	0.0296	0.0744	0.0094
0.4865	0.0489	0.0397	0.1141	0.0079
0.5001	0.0709	0.0633	0.1408	0.0044
0.5063	0.0348	0.0173	0.0563	0.0194
0.5112	0.1344	0.0853	0.2577	0.0166
0.5113	0.0418	0.0198	0.0743	0.0224
0.5244	0.0410	0.0545	0.1366	0.0067
0.5260	0.0425	0.0172	0.0610	0.0240
0.5451	0.0555	0.0658	0.1717	0.0149
0.5465	0.0193	0.0288	0.0698	0.0005
0.5519	0.0662	0.0780	0.1802	0.0004
0.5677	0.0552	0.0332	0.0932	0.0102
0.5778	0.0822	0.0626	0.1657	0.0127
0.5910	0.1112	0.0355	0.1583	0.0806
0.6219	0.0460	0.0242	0.0719	0.0155
0.6301	0.0774	0.1135	0.2753	0.0003
0.6482	0.0473	0.0462	0.1032	0.0009
0.6610	0.0534	0.0439	0.1032	0.0065
0.6624	0.0347	0.0158	0.0564	0.0184
0.6634	0.0824	0.0841	0.2266	0.0146
0.7107	0.0397	0.0393	0.1045	0.0042
0.7779	0.0765	0.0526	0.1291	0.0033
0.7954	0.0384	0.0331	0.0936	0.0082
0.8031	0.0904	0.0774	0.1834	0.0067
0.8284	0.1032	0.0746	0.2125	0.0320
0.8290	0.0603	0.0325	0.0934	0.0251
0.8431	0.0484	0.0382	0.1089	0.0140
0.8574	0.0202	0.0199	0.0554	0.0093
0.8588	0.0479	0.0216	0.0771	0.0198
0.9208	0.0459	0.0247	0.0749	0.0218
0.9312	0.1099	0.1208	0.3141	0.0232

Appendix C

Variance Analysis for the Primary Care Model

One way ANOVA, computed with OpenStat.

Variable labels for the group variables (variable name is scenario):
1 - BR-3/FFS
2 - BR-3/Capitation
3 - BR-6/FFS
4 - BR-6/Capitation
5 - RL-3/FFS
6 - RL-3/Capitation
7 - RL-6/FFS
8 - RL-6/Capitation

```
=====================================================================
ONE WAY ANALYSIS OF VARIANCE RESULTS

Dependent variable is: wait, Independent variable is: scenario

---------------------------------------------------------------------
SOURCE    D.F.      SS        MS        F         PROB.>F   OMEGA SQR.
---------------------------------------------------------------------
BETWEEN      7   4144.04    592.01     29.74      0.00      0.30
WITHIN     458   9117.10     19.91
TOTAL      465  13261.14
---------------------------------------------------------------------
```

MEANS AND VARIABILITY OF THE DEPENDENT VARIABLE FOR LEVELS OF THE INDEPENDENT VARIABLE

GROUP	MEAN	VARIANCE	STD.DEV.	N
1	9.63	32.61	5.71	60
2	13.94	30.45	5.52	60
3	11.64	16.91	4.11	60
4	14.07	36.44	6.04	60
5	6.65	6.31	2.51	60
6	9.22	15.94	3.99	60
7	5.15	5.31	2.30	60
8	8.81	12.13	3.48	60
TOTAL	9.98	28.52	5.34	466

TESTS FOR HOMOGENEITY OF VARIANCE

Hartley Fmax test statistic = 6.86 with deg.s freem: 8 and 59.
Cochran C statistic = 0.23 with deg.s freem: 8 and 59.
Bartlett Chi-square = 91.64 with 7 D.F. Prob. > Chi-Square = 0.000

FISHER'S (PROTECTED) LEAST SIGNIFICANT DIFFERENCE TEST

GROUP	MEAN	GROUP	MEAN	DIFFERENCE	FISHER LSD	SIGNIFICANT?
1	9.626	2	13.944	4.319	1.601	YES
1	9.626	3	11.636	2.010	1.601	YES
1	9.626	4	14.070	4.445	1.601	YES
1	9.626	5	6.652	2.973	1.698	YES
1	9.626	6	9.224	0.402	1.601	NO
1	9.626	7	5.146	4.480	1.608	YES
1	9.626	8	8.810	0.815	1.608	NO
2	13.944	3	11.636	2.308	1.601	YES
2	13.944	4	14.070	0.126	1.601	NO
2	13.944	5	6.652	7.292	1.698	YES
2	13.944	6	9.224	4.720	1.601	YES
2	13.944	7	5.146	8.798	1.608	YES
2	13.944	8	8.810	5.134	1.608	YES
3	11.636	4	14.070	2.434	1.601	YES
3	11.636	5	6.652	4.984	1.698	YES
3	11.636	6	9.224	2.412	1.601	YES
3	11.636	7	5.146	6.490	1.608	YES
3	11.636	8	8.810	2.826	1.608	YES
4	14.070	5	6.652	7.418	1.698	YES
4	14.070	6	9.224	4.846	1.601	YES
4	14.070	7	5.146	8.924	1.608	YES
4	14.070	8	8.810	5.260	1.608	YES
5	6.652	6	9.224	2.572	1.698	YES
5	6.652	7	5.146	1.507	1.704	NO
5	6.652	8	8.810	2.158	1.704	YES
6	9.224	7	5.146	4.078	1.608	YES
6	9.224	8	8.810	0.414	1.608	NO
7	5.146	8	8.810	3.665	1.614	YES

NOTE! Familywise error rate may be greater than alpha

ONE WAY ANALYSIS OF VARIANCE RESULTS

Dependent variable is: effort , Independent variable is: scenario

SOURCE	D.F.	SS	MS	F	PROB.>F	OMEGA SQR.
BETWEEN	7	0.04	0.01	59.99	0.00	0.47
WITHIN	458	0.05	0.00			
TOTAL	465	0.09				

MEANS AND VARIABILITY OF THE DEPENDENT VARIABLE FOR LEVELS OF THE INDEPENDENT VARIABLE

GROUP	MEAN	VARIANCE	STD.DEV.	N
1	0.98	0.00	0.00	60
2	0.98	0.00	0.01	60
3	0.98	0.00	0.00	60
4	0.98	0.00	0.01	60
5	0.96	0.00	0.02	60
6	0.96	0.00	0.01	60
7	0.96	0.00	0.01	60
8	0.95	0.00	0.02	60
TOTAL	0.97	0.00	0.01	466

TESTS FOR HOMOGENEITY OF VARIANCE

Hartley Fmax test statistic = 115294120037.52 with deg.s freem: 8 and 59.
Cochran C statistic = 0.31 with deg.s freem: 8 and 59.
Bartlett Chi-square = 1608.00 with 7 D.F. Prob. > Chi-Square = 0.001

FISHER'S (PROTECTED) LEAST SIGNIFICANT DIFFERENCE TEST

GROUP	MEAN	GROUP	MEAN	DIFFERENCE	FISHER LSD	SIGNIFICANT?
1	0.980	2	0.976	0.004	0.004	NO
1	0.980	3	0.980	0.000	0.004	NO
1	0.980	4	0.976	0.004	0.004	YES
1	0.980	5	0.963	0.017	0.004	YES
1	0.980	6	0.960	0.020	0.004	YES
1	0.980	7	0.960	0.020	0.004	YES
1	0.980	8	0.955	0.025	0.004	YES
2	0.976	3	0.980	0.004	0.004	YES
2	0.976	4	0.976	0.000	0.004	NO
2	0.976	5	0.963	0.013	0.004	YES
2	0.976	6	0.960	0.016	0.004	YES
2	0.976	7	0.960	0.017	0.004	YES
2	0.976	8	0.955	0.021	0.004	YES
3	0.980	4	0.976	0.004	0.004	YES
3	0.980	5	0.963	0.017	0.004	YES
3	0.980	6	0.960	0.020	0.004	YES
3	0.980	7	0.960	0.021	0.004	YES
3	0.980	8	0.955	0.025	0.004	YES
4	0.976	5	0.963	0.013	0.004	YES
4	0.976	6	0.960	0.016	0.004	YES
4	0.976	7	0.960	0.016	0.004	YES
4	0.976	8	0.955	0.021	0.004	YES

```
  5     0.963   6     0.960     0.003     0.004         NO
  5     0.963   7     0.960     0.003     0.004         NO
  5     0.963   8     0.955     0.008     0.004         YES
  6     0.960   7     0.960     0.001     0.004         NO
  6     0.960   8     0.955     0.005     0.004         YES
  7     0.960   8     0.955     0.005     0.004         YES
NOTE! Familywise error rate may be greater than alpha
------------------------------------------------------------------------
========================================================================


ONE WAY ANALYSIS OF VARIANCE RESULTS

Dependent variable is: referral_rate, Independent variable is: scenario

------------------------------------------------------------------------
SOURCE    D.F.      SS        MS        F         PROB.>F   OMEGA SQR.
------------------------------------------------------------------------
BETWEEN      7      3.46      0.49    884.98      0.00      0.93
WITHIN     458      0.26      0.00
TOTAL      465      3.71
------------------------------------------------------------------------

MEANS AND VARIABILITY OF THE DEPENDENT VARIABLE FOR LEVELS OF THE INDEPENDENT VARIABLE
------------------------------------------------------------------------
GROUP     MEAN      VARIANCE  STD.DEV.  N
------------------------------------------------------------------------
   1       0.81      0.00      0.02  60
   2       0.99      0.00      0.04  60
   3       0.91      0.00      0.02  60
   4       0.99      0.00      0.03  60
   5       0.80      0.00      0.01  60
   6       0.99      0.00      0.02  60
   7       0.81      0.00      0.02  60
   8       1.00      0.00      0.02  60
------------------------------------------------------------------------
TOTAL      0.92      0.01      0.09 466
------------------------------------------------------------------------

TESTS FOR HOMOGENEITY OF VARIANCE
------------------------------------------------------------------------
Hartley Fmax test statistic =       7.24 with deg.s freem: 8 and    59.
Cochran C statistic =       0.28 with deg.s freem: 8 and    59.
Bartlett Chi-square =      70.53 with    7 D.F. Prob. > Chi-Square =  0.000
------------------------------------------------------------------------

FISHER'S (PROTECTED) LEAST SIGNIFICANT DIFFERENCE TEST
------------------------------------------------------------------------
GROUP   MEAN  GROUP   MEAN   DIFFERENCE  FISHER LSD  SIGNIFICANT?
------------------------------------------------------------------------
  1     0.808   2     0.988     0.180     0.008         YES
  1     0.808   3     0.910     0.102     0.008         YES
  1     0.808   4     0.991     0.183     0.008         YES
  1     0.808   5     0.804     0.004     0.009         NO
  1     0.808   6     0.995     0.187     0.008         YES
  1     0.808   7     0.805     0.003     0.009         NO
  1     0.808   8     0.999     0.191     0.009         YES
  2     0.988   3     0.910     0.078     0.008         YES
  2     0.988   4     0.991     0.003     0.008         NO
  2     0.988   5     0.804     0.184     0.009         YES
  2     0.988   6     0.995     0.007     0.008         NO
```

```
  2     0.988   7     0.805     0.183     0.009         YES
  2     0.988   8     0.999     0.011     0.009         YES
  3     0.910   4     0.991     0.081     0.008         YES
  3     0.910   5     0.804     0.107     0.009         YES
  3     0.910   6     0.995     0.085     0.008         YES
  3     0.910   7     0.805     0.105     0.009         YES
  3     0.910   8     0.999     0.089     0.009         YES
  4     0.991   5     0.804     0.187     0.009         YES
  4     0.991   6     0.995     0.004     0.008         NO
  4     0.991   7     0.805     0.186     0.009         YES
  4     0.991   8     0.999     0.008     0.009         NO
  5     0.804   6     0.995     0.191     0.009         YES
  5     0.804   7     0.805     0.002     0.009         NO
  5     0.804   8     0.999     0.195     0.009         YES
  6     0.995   7     0.805     0.189     0.009         YES
  6     0.995   8     0.999     0.004     0.009         NO
  7     0.805   8     0.999     0.194     0.009         YES
NOTE! Familywise error rate may be greater than alpha
------------------------------------------------------------------------
========================================================================


ONE WAY ANALYSIS OF VARIANCE RESULTS

Dependent variable is: utility_1, Independent variable is: scenario

------------------------------------------------------------------------
SOURCE    D.F.      SS        MS        F          PROB.>F   OMEGA SQR.
------------------------------------------------------------------------
BETWEEN      7    152.53     21.79   1670.86      0.00      0.59
WITHIN    7992    104.23      0.01
TOTAL     7999    256.76
------------------------------------------------------------------------

MEANS AND VARIABILITY OF THE DEPENDENT VARIABLE FOR LEVELS OF THE INDEPENDENT VARIABLE
------------------------------------------------------------------------
GROUP     MEAN      VARIANCE  STD.DEV.  N
------------------------------------------------------------------------
   1       0.62      0.00      0.05    1000
   2       0.61      0.00      0.05    1000
   3       0.64      0.00      0.03    1000
   4       0.61      0.00      0.04    1000
   5       0.44      0.01      0.11    1000
   6       0.29      0.02      0.16    1000
   7       0.45      0.02      0.16    1000
   8       0.29      0.04      0.19    1000
------------------------------------------------------------------------
TOTAL      0.49      0.03      0.188000
------------------------------------------------------------------------

TESTS FOR HOMOGENEITY OF VARIANCE
------------------------------------------------------------------------
Hartley Fmax test statistic =      47.98 with deg.s freem: 8 and   999.
Cochran C statistic =       0.34 with deg.s freem: 8 and   999.
Bartlett Chi-square =    5811.61 with    7 D.F. Prob. > Chi-Square =  0.001
------------------------------------------------------------------------
```

FISHER'S (PROTECTED) LEAST SIGNIFICANT DIFFERENCE TEST

GROUP	MEAN	GROUP	MEAN	DIFFERENCE	FISHER LSD	SIGNIFICANT?
1	0.616	2	0.611	0.006	0.010	NO
1	0.616	3	0.638	0.021	0.010	YES
1	0.616	4	0.612	0.005	0.010	NO
1	0.616	5	0.440	0.177	0.010	YES
1	0.616	6	0.289	0.327	0.010	YES
1	0.616	7	0.454	0.162	0.010	YES
1	0.616	8	0.287	0.330	0.010	YES
2	0.611	3	0.638	0.027	0.010	YES
2	0.611	4	0.612	0.001	0.010	NO
2	0.611	5	0.440	0.171	0.010	YES
2	0.611	6	0.289	0.322	0.010	YES
2	0.611	7	0.454	0.157	0.010	YES
2	0.611	8	0.287	0.324	0.010	YES
3	0.638	4	0.612	0.026	0.010	YES
3	0.638	5	0.440	0.198	0.010	YES
3	0.638	6	0.289	0.348	0.010	YES
3	0.638	7	0.454	0.183	0.010	YES
3	0.638	8	0.287	0.351	0.010	YES
4	0.612	5	0.440	0.172	0.010	YES
4	0.612	6	0.289	0.323	0.010	YES
4	0.612	7	0.454	0.158	0.010	YES
4	0.612	8	0.287	0.325	0.010	YES
5	0.440	6	0.289	0.151	0.010	YES
5	0.440	7	0.454	0.014	0.010	YES
5	0.440	8	0.287	0.153	0.010	YES
6	0.289	7	0.454	0.165	0.010	YES
6	0.289	8	0.287	0.003	0.010	NO
7	0.454	8	0.287	0.168	0.010	YES

NOTE! Familywise error rate may be greater than alpha

Appendix D

Software and Models

Table D.1: Executable models

Name	Description	URL	Date
gsim	JBoss server with gsim.jar and models (optimised for windows platform)	http://www.stephan-schuster.net/gsim/framework/gsim-windows-0.1.zip	09/09/2010
DynamicProcess	Utility used for computing stochastic stable networks (jar library)	http://www.stephan-schuster.net/gsim/DynamicProcess/dp.jar	09/09/2010
PrimaryCare	The primary care model	http://www.stephan-schuster.net/gsim/models/PrimaryCare/gp.zip	09/09/2010
Networks-1	The network base model	http://www.stephan-schuster.net/gsim/models/Networks1/net-base.zip	20/06/2011
Networks-2	The network BRA model	http://www.stephan-schuster.net/gsim/models/Networks2/net-bra.zip	09/09/2010
Discrimination	The discrimination model	http://www.stephan-schuster.net/models/Discrimination/discrimination.zip	05/01/2012

Table D.2: Source code

Name	Description	URL	Date
gsim	Sources of the gsim framework	http://www.stephan-schuster.net/gsim/framework/src.zip	09/09/2010
DynamicProcess	Sources of the DynamicProcess jar-file	http://www.stephan-schuster.net/gsim/DynamicProcess/src.zip	09/09/2010
PrimaryCare	Source of the primary care model	http://www.stephan-schuster.net/gsim/PrimaryCare/src.zip	09/09/2010
Networks-1	Sources of the network base model	http://www.stephan-schuster.net/gsim/models/Networks1/src.zip	20/06/2011
Networks-2	Sources of the network BRA model	http://www.stephan-schuster.net/gsim/models/Networks2/src.zip	09/09/2010
Discrimination	Sources of the discrimination model	http://www.stephan-schuster.net/gsim/models/Discrimination/src.zip	05/01/2012